YAMAHA

DIGITAL PROGRAMMABLE
ALGORITHM SYNTHESIZER

DX7

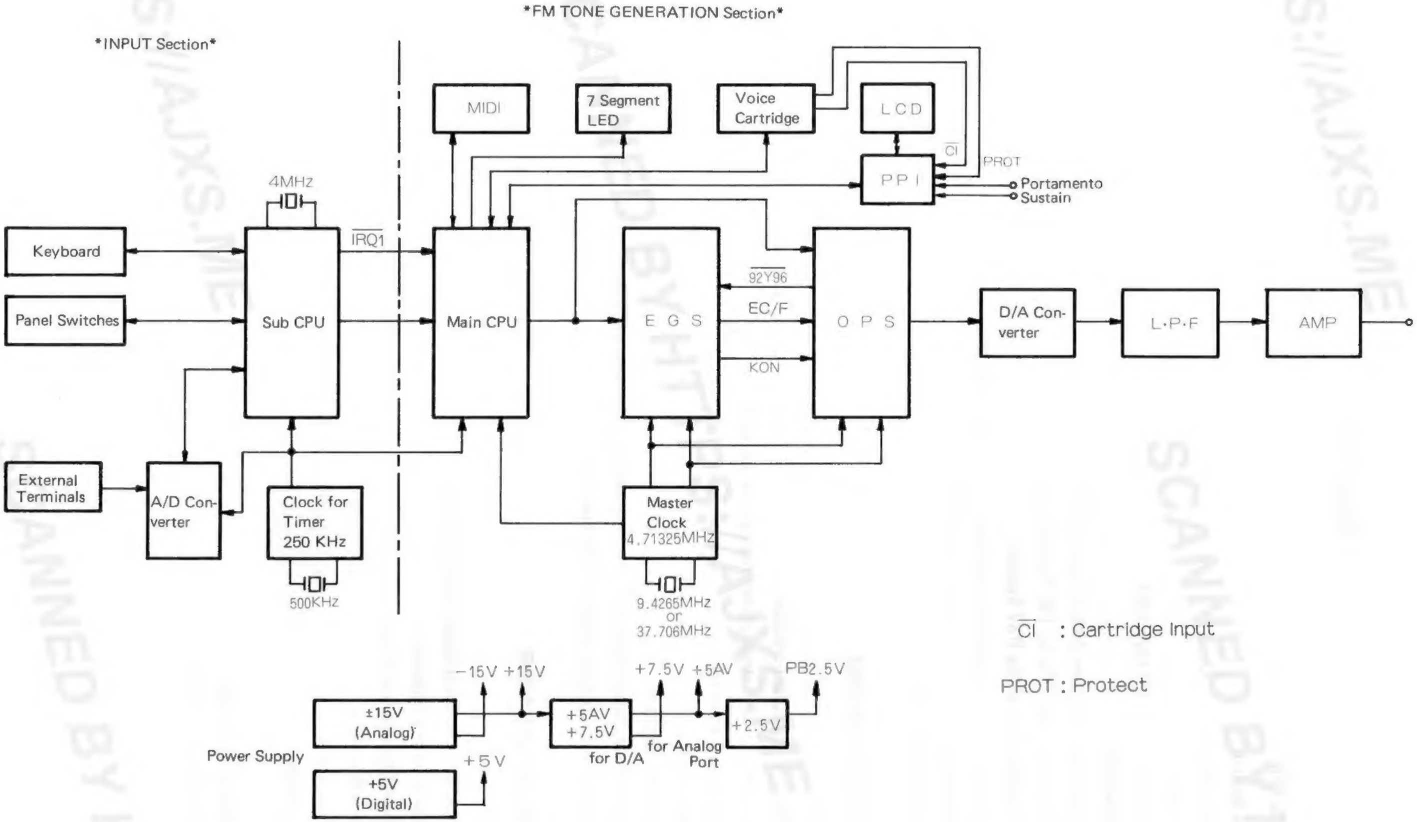
TECHNICAL ANALYSIS

006639

SINCE 1887 YAMAHA
NIPPON GAKKI CO., LTD. HAMAMATSU, JAPAN
1.0K-426 Ⓢ Printed in Japan '84.10

CONTENTS

DX7 Block Diagram



$\overline{CI}$: Cartridge Input

PROT : Protect

Fig 1.

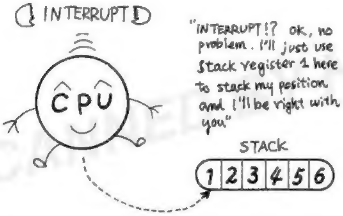
REMARKS

The remarks and comments listed on the left hand side of the text may be helpful in explaining a term or concept relating to the text.

What is a stack ?

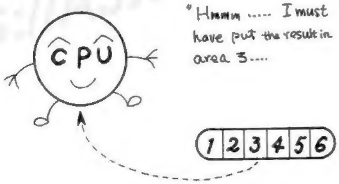
The CPU must stop running a routine when it is interrupted or when there is a call to run a subroutine. When it finishes with the interrupt of sub routine it then needs to return to the main routine, but how does it remember where it stopped? The CPU must write its position at times like these into a RAM. Each time it is interrupted or called it "stacks" this position in the RAM so that it can return to it.

The area of the RAM that is used for this operation is called the "stack." The CPU also has "stack pointer" that is used to direct it to the position laid on the stack.


INTERRUPT
CPU
"INTERRUPT!? OK, no problem. I'll just use Stack register 1 here to stack my position and I'll be right with you"
STACK
1 2 3 4 5 6

How about the work area ?

The registers of the CPU itself are called "internal registers," and those outside the CPU are called "external registers." The work area is the in the RAM, and is allotted to make up for the lack of internal registers. The work area is used to store the results of computations temporally as well as "flags" which are used to show the status of the external memory.



I THE DX7 SYSTEM

The circuits of the DX7 are shown in their basic form in figure (1); the hardware consists of two basic groups which are not synchronized to each other.

- **Input/Read Section**
- **FM Tone Generation Section** (including MIDI, voice cartridge, liquid crystal display, 7 segment LED)

(1) Input/Read Section

1) Sub CPU (6805S)

This CPU is a microcomputer unit that has:

- **a 64 byte RAM** (for stack and work area)
- **a 1100 byte ROM** (mask ROM, prepared by IC maker)
- **I/O ports**
- **a timer**

These features are integrated internally, there is no external ROM nor RAM.

The 1100 byte ROM contains a program so that the CPU can function as a key scanner. The sub CPU reads data periodically (scans) from the:

- **Keyboard**
- **External Terminals (DATA ENTRY, Modulation Wheel, Pitch Bend Wheel, etc.)**
- **Panel Switches**

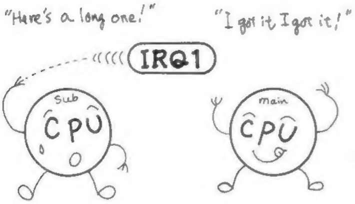
2) Scan Cycle Time

The scan cycle time is the time between sub CPU readings from the sections shown above. The times shown below are on the condition that the keyboard and panel are not touched.

* **Keyboard**	: **500 μsec**
* **External Terminals**	: **4.5 msec**
* **Panel Switches**	: **27 msec**

REMARKS

Don't forget that terminal IRQ1 is not the only one that accepts interruptions, $\overline{\text{IRQ2}}$ is used for internal signals and it is used for MIDI.



3) Transmitting Data to the Main CPU

All data from the sub CPU to the main CPU is transmitted through interrupt routines.

When an event occurs on the keyboard, panel switches, etc., the sub CPU receives that data and interrupts the main CPU through the IRQ1 terminal. Once the main CPU has accepted interruption, the data is transmitted from terminals $A_0 \sim AA_7$ of the sub CPU to terminals $P_{10} \sim P_{17}$ of the main CPU. When the data transmission has been completed, the main CPU sends an ACCEPT signal to the sub CPU; if there is any more data to be sent from the sub CPU, the ACCEPT signal tells the sub CPU that the main CPU is ready for its transmission.

REMARKS

OK, got all your data here for you, time to arithmetically calculate frequency modulation!
EGS
OPS
Thats easy for you to say, all you have to do is COLLECT the data ...

(2) FM Tone Generation

The FM tone Generator is the heart of the DX7; it is composed of the EGS and the OPS.

What is the EGS ?

"EGS" is short for "the Envelope Generator for the Synthesizer." It is an LSI that produces (generates) the digital data for the frequency information and the envelope waveform, depending upon various data and parameters from the main CPU. It produces this digital data according to the rate, level, key code, and other voicing data that it also receives from the main CPU. This data is synchronized with the signal (92Y96 sync pulse) for synchronization of the system and sent to $EC_1 \sim EC_{12}$ of the OPS in the case of volume envelope data, and $F_1 \sim F_{14}$ in the case of frequency data.

What is the OPS ?

"OPS" stands for "the Operator for the Synthesizer." It is an LSI that performs FM Arithmetic Operation, with the data transmitted from the EGS, and the result is the final tone signal in digital data. This digital data is transmitted to the D/A converter.
The OPS has the equivalent of 16 notes polyphonic, and can simulate as many as 6 oscillators per note depending on what algorithm is selected. The ALGORITHM decides how the wiring is to be done.

1) The Relationship Between the EGS and the OPS

The EGS is master (main) and the OPS is slave (sub). The OPS uses the ALGORITHM already set within it to arithmetically operate on the frequency modulation (FM) for the data in the FM arithmetic operation parameters: $EC_1 \sim EC_{12}$, $F_1 \sim F_{14}$.

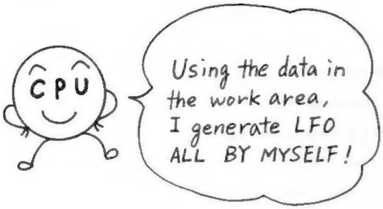
2) The Role of the Main CPU for FM Tone Generation

The Main CPU: 63B03, is a microcomputing unit that includes:

- **a 128 byte RAM**
- **13 I/O ports**
- **a 16 bit timer**
- **SCI (Serial Communication Interface)**

There is NO circuit anywhere in the unit for the generation of LFO.

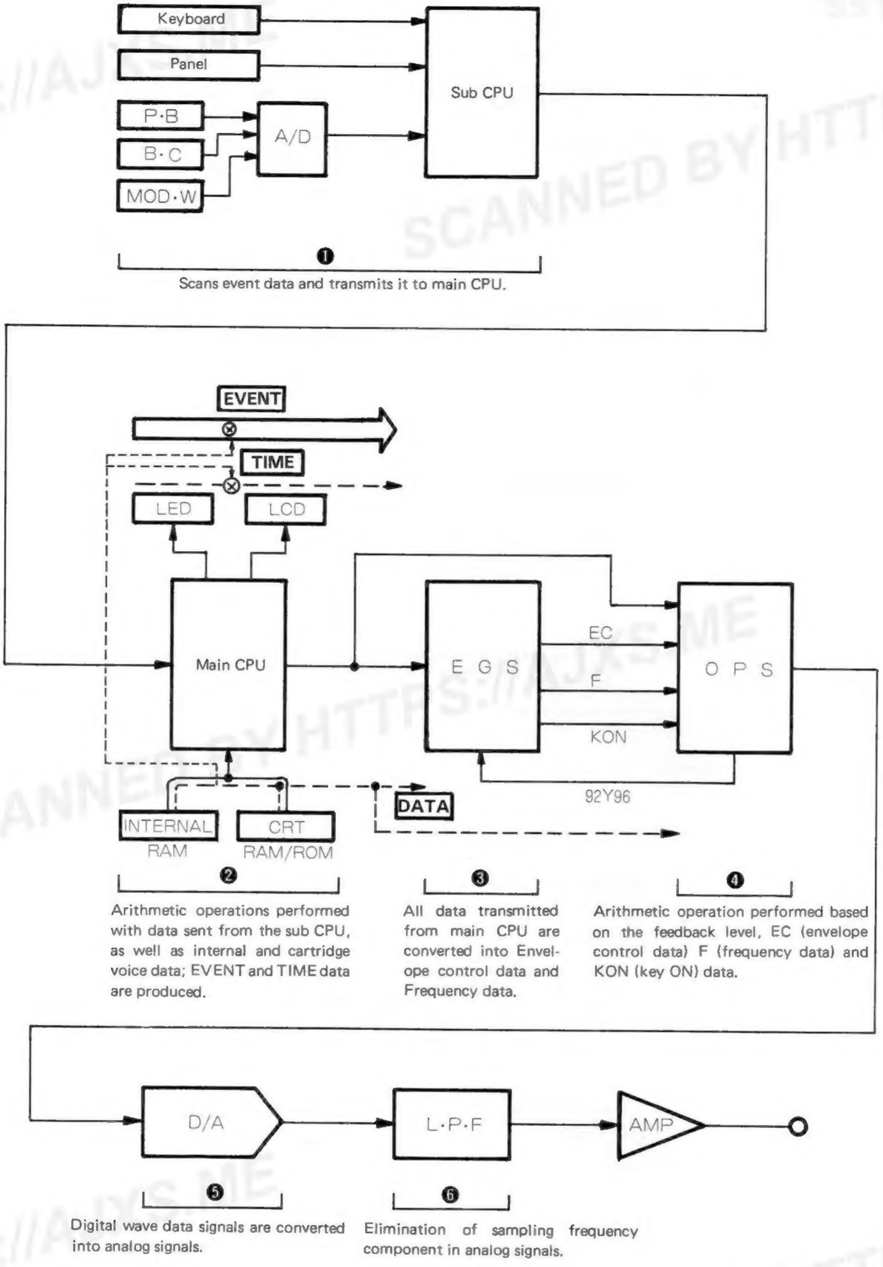
REMARKS


CPU
Using the data in the work area, I generate LFO ALL BY MYSELF!

In the DX7 function mode, the input of LFO related data is done by writing it in the work area in the RAM (IC21). Of course, the stack area is also in the RAM (IC21).

Usually, conventional keyboards use a CPU for the control of voicing data. But since the tone signal is converted into digital, the main CPU of the DX7 generates or modifies all the data needed for the tone signal to be frequency modulated by itself. Those data are then transmitted to the EGS and the OPS. The easiest to understand example of such operations is the CPU internal arithmetic operation to generate the LFO waveform.

NOTES



❶ Scans event data and transmits it to main CPU.

❷ Arithmetic operations performed with data sent from the sub CPU, as well as internal and cartridge voice data; EVENT and TIME data are produced.

❸ All data transmitted from main CPU are converted into Envelope control data and Frequency data.

❹ Arithmetic operation performed based on the feedback level, EC (envelope control data) F (frequency data) and KON (key ON) data.

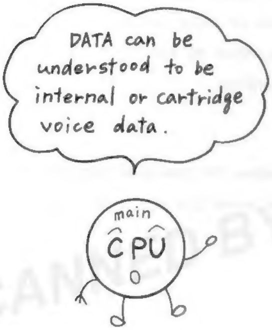
❺ Digital wave data signals are converted into analog signals.

❻ Elimination of sampling frequency component in analog signals.

Figure 2

REMARKS

An EVENT can be understood as things that happen on the keyboard.
main
CPU


DATA can be understood to be internal or cartridge voice data.
main
CPU

(3) Outline of the Operation of the DX7 System

1) Classification of Data Transmitted from Main CPU to EGS, and OPS

(Understanding the roles of the Main CPU Better)

In order to understand how the DX7 works, it is good to know what kind of data it uses. When we consider their nature and content, we can divide them up into 3 types of data:

- **EVENT**
- **DATA**
- **TIME**

NOTE: These three special data names have been given for special purposes. Do not confuse them with terms used in a more general sense.

- **EVENT :** This data changes while the instrument is being played, and is sent to the EGS. For example: key codes, Key ON/OFF, Initial Touch, Transposition, Pitch EG etc.
- **DATA :** This data does not need to change while the instrument is played. It is fixed data. When the voice button is chosen, this data is sent to be registered in the internal registers of the EGS and OPS. For example: EG-RATE, EG-LEVEL, DETUNE, COARSE, FINE, ALGORITHM, FEED BACK LEVEL, etc.
- **TIME :** This data must be transmitted to the EGS periodically while event data is being generated. For example: LFO, BREATH CONTROL, PITCH BEND wheel, etc.

2) DX7 System Operation

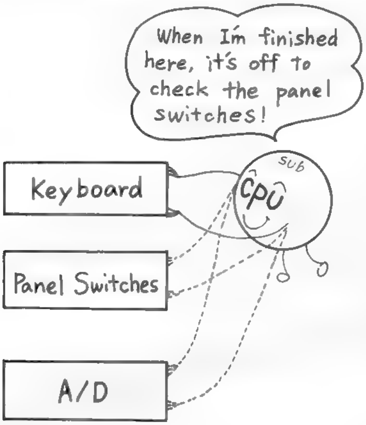
The DX7 system has 4 different modes:

- **PLAY**
- **EDIT**
- **FUNCTION**
- **STORE**

The EDIT, FUNCTION and STORE modes are basically for editing and adding data to the RAM work area (IC21), so here, we will discuss operation during the PLAY mode.

REMARKS

It's the job of the sub CPU to scan all the data from the input sections.


When I'm finished here, it's off to check the panel switches!
Keyboard
Panel Switches
A/D
sub
CPU

Look at figure 2 and you will see a series of numbers, follow these numbers as you read the following explanation:

1. The sub CPU constantly scans the input sections to detect events, and when there is an event, it interrupts the main CPU and transmits that new data. It is 2 bytes of data; and it is composed in the following manner:

   Byte 1 : Data Identifier (describes type of data)
   For example keyboard data identifies the note played on the keyboard.

   Byte 2 : Data(actual data of the value or magnitude of a control, etc.)
   Initial touch data of keyboard is included.

2. The main CPU reacts to the data transmitted from the sub CPU in the following ways:

   (1) A Panel Switch is pressed

   When this happens, the DATA related to the switch's code number is taken from the internal RAM or voice cartridge and sent to the EGS and OPS via interrupt routines to the main CPU. Then, the OPS will have received data regarding:

   - **ALGORITHM No.**
   - **FEEDBACK LEVEL**

   In order to let the user know about the change in data, the change of data from the internal RAM or voice cartridge is displayed with the LED and LCD displays.

   (2) Keyboard ON

   When the keyboard is on:

   Ⓐ * The key code data is taken from the sub CPU.
   * Transpose
   * Pitch EC } Taken from internal or voice cartridge memory

   Then, this data is processed by the main CPU, and transmitted to the EGS as a –

   - **Frequency EVENT**

   Ⓑ * Initial Touch data is taken from the sub CPU
   * OP output level
   * Level Scaling } Taken from internal or voice cartridge memory

   Then, this data is processed by the main CPU, and transmitted to the EGS as a –

   - **Level EVENT**

   C Key ON and key OFF channels are distinguished from each other and transmitted to the EGS as a –

   - **Key ON/OFF EVENT**

(3) LFO and Modulation Data Operations

Ⓐ * After Touch } From sub CPU or RAM work area
* Breath Control }

* LFO ———— From internal or voice cartridge memory

The above data is processed in the main CPU, and transmitted to the EGS as –

* **Amplitude Modulation TIME**

Ⓑ * Pitch Bend —— From sub CPU and RAM work area

* LFO ———— From Internal or voice cartridge memory

The above data is processed in the main CPU, and transmitted to the EGS as –

* **Pitch Modulation TIME**

3. The EGS integrates the parameters, and the corresponding EVENT and TIME data, and changes it to –

* **EC: Volume Envelope Control Data**

* **F : Frequency Data**

and with the

* **KON : Key ON Data**

These are sent to the OPS, synchronized with the 92Y96 synchronization signal.

4. The OPS receives from the main CPU –

* ALGORITHM NO.

* FEEDBACK LEVEL

for each operator; and performs FM arithmetic operations based on the EC, F, and KON data synchronized with 92Y96 for each channel, and generates 16 channels of digital data at the most.

5. The data input to the D/A converter from the OPS is converted into analog waveforms, but at this point, a signal consisting of waveforms layered with the sampling frequency is what is generated.

6. It is only after passing through the LPF that the sampling frequency is filtered out, and the pure analog signal is gained.

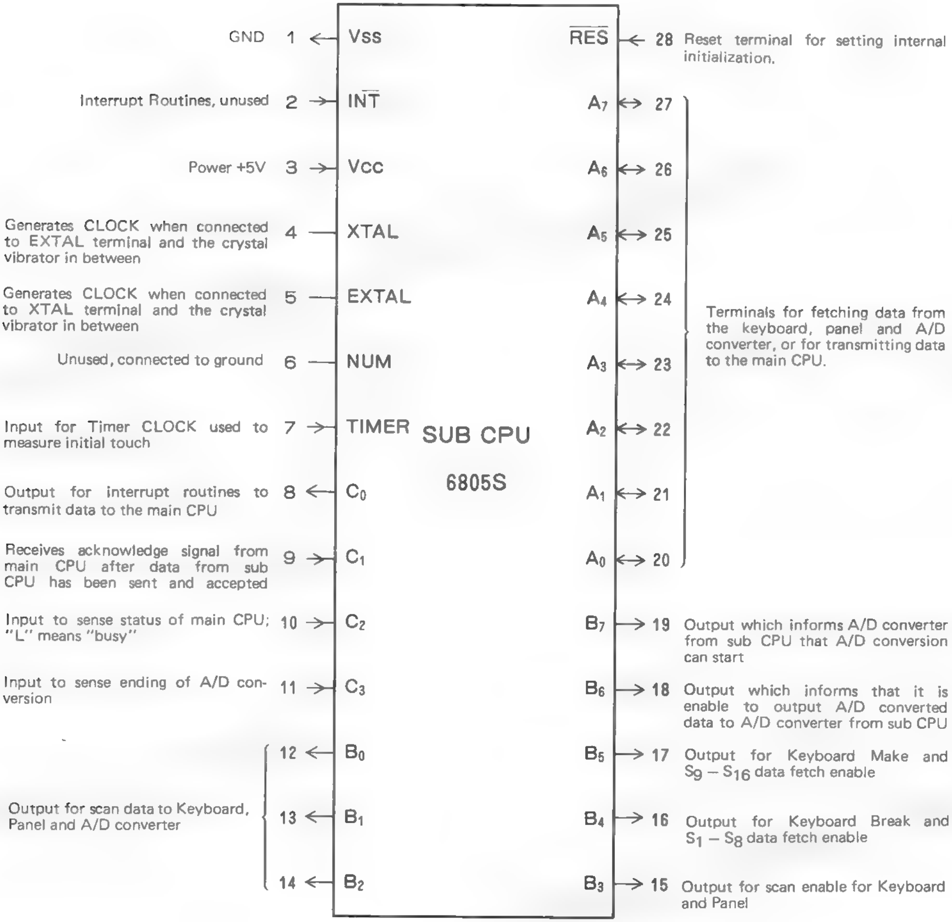
Ⅱ THE CIRCUITS OF THE DX7

(1) Input Sections

1) Sub CPU Terminals and Functions

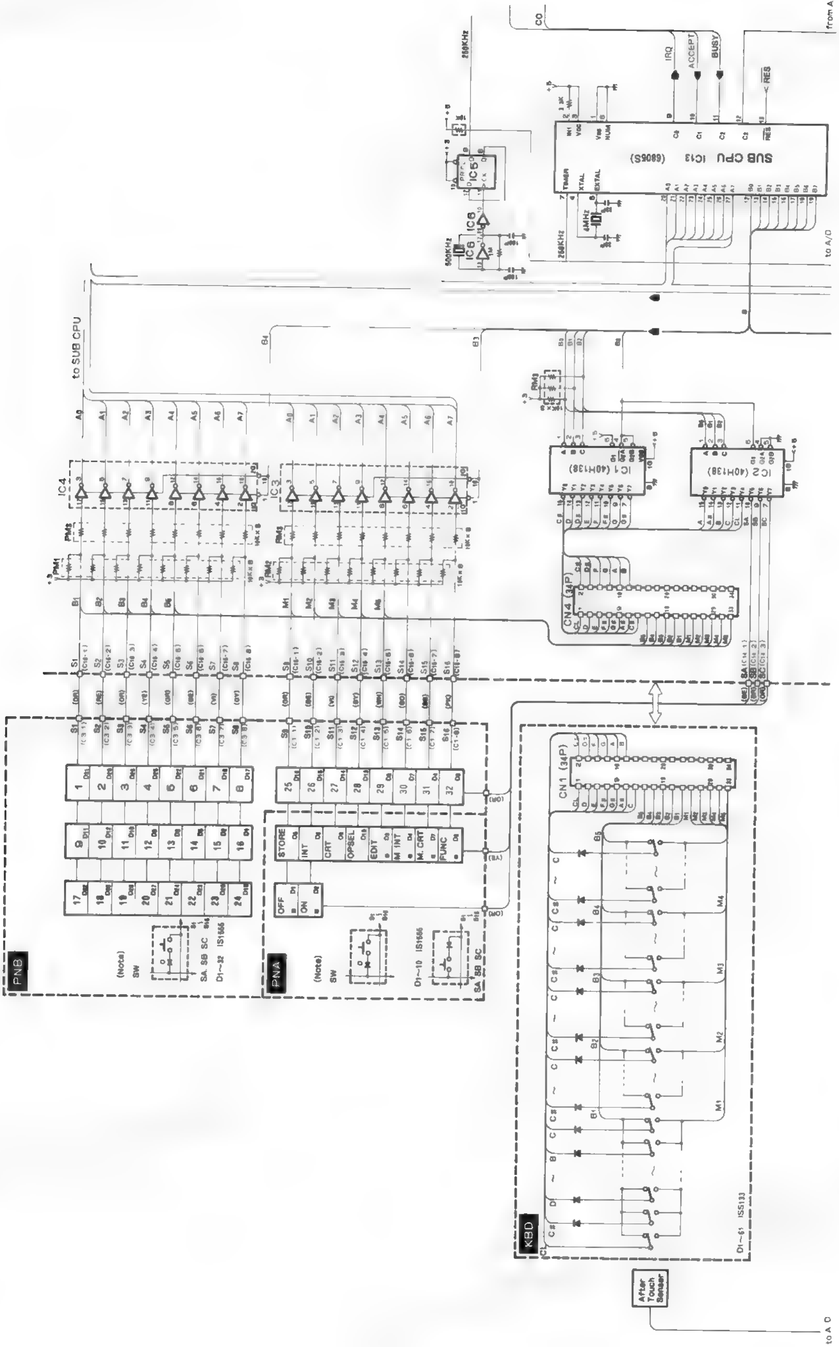
See figures I–1 ~ I–6 (Pages 64, 65)

The function of the sub CPU is determined by the program that has been burnt into its internal ROM; below, there is a diagram which explains in more detail. This CPU unit's basic purpose is to scan the various I/O, and transmit the data gained to the main CPU with interrupt routines.



SUB CPU
6805S
GND 1 Vss
Interrupt Routines, unused 2 INT
Power +5V 3 Vcc
Generates CLOCK when connected to EXTAL terminal and the crystal vibrator in between 4 XTAL
Generates CLOCK when connected to XTAL terminal and the crystal vibrator in between 5 EXTAL
Unused, connected to ground 6 NUM
Input for Timer CLOCK used to measure initial touch 7 TIMER
Output for interrupt routines to transmit data to the main CPU 8 C0
Receives acknowledge signal from main CPU after data from sub CPU has been sent and accepted 9 C1
Input to sense status of main CPU; "L" means "busy" 10 C2
Input to sense ending of A/D conversion 11 C3
12 B0
13 B1
14 B2
Output for scan data to Keyboard, Panel and A/D converter
RES 28 Reset terminal for setting internal initialization.
A7 27
A6 26
A5 25
A4 24
A3 23
A2 22
A1 21
A0 20
Terminals for fetching data from the keyboard, panel and A/D converter, or for transmitting data to the main CPU.
B7 19 Output which informs A/D converter from sub CPU that A/D conversion can start
B6 18 Output which informs that it is enable to output A/D converted data to A/D converter from sub CPU
B5 17 Output for Keyboard Make and S9 – S16 data fetch enable
B4 16 Output for Keyboard Break and S1 – S8 data fetch enable
B3 15 Output for scan enable for Keyboard and Panel

Keyboard and Panel Switch Circuit



REMARKS

Remember initial touch will not be sensed; the highest level will be assumed, and the voice will have less variation unless the sub CPU is the only unit that has the input of 250KHz.

Refer to page 14.

Panel Switch Codes

1	50H	22	65H
2	51H	23	66H
3	52H	24	67H
4	53H	25	68H
5	54H	26	69H
6	55H	27	6AH
7	56H	28	6BH
8	57H	29	6CH
9	58H	30	6DH
10	59H	31	6EH
11	5AH	32	6FH
12	5BH	STORE	70H
13	5CH	PROTECT INT	71H
14	5DH	CRT	72H
15	5EH	OPSEL	73H
16	5FH	EDIT	74H
17	60H	MEMORY INT	75H
18	61H	CRT	76H
19	62H	FUNCTION	77H
20	63H	OFF	78H
21	64H	OH	79H

2) Codes for the different types of data used by the sub CPU

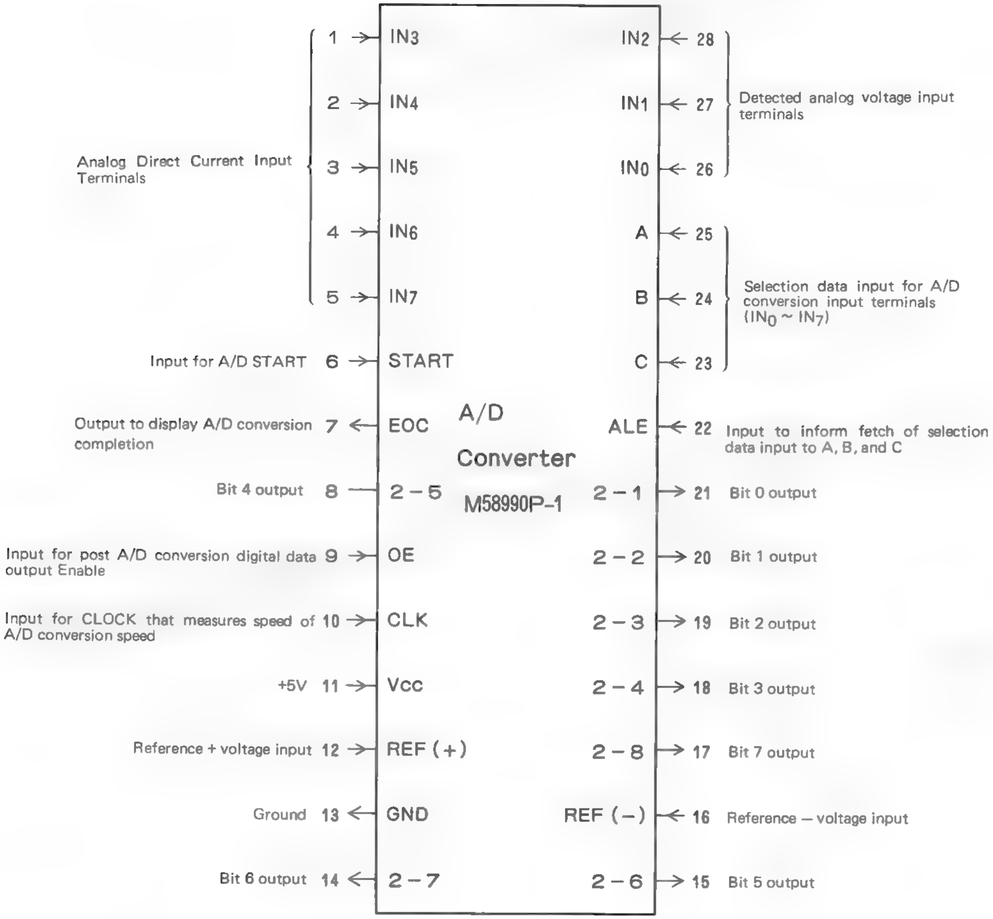
The sub CPU transmits data composed of two bytes to the main CPU. The table below is a list and description of this data. Knowledge of the composition of the data itself is not required for the servicing of the unit, but it does help to know about it in order to understand the function of the sub CPU.

		Byte 1	Byte 2
		Identifier	Data
Panel Switch	ON	98H	Switch is ON
	OFF	99H	Switch is OFF
A/D	DATA ENTRY	90H	Values for 00H～7FH
	PITCH BENDER	91H	
	MODULATION WHEEL	92H	
	FOOT CONTROLLER	93H	
	BREATH CONTROLLER	94H	
	AFTER TOUCH	95H	
	BATTERY	96H	
	Unused	97H	
Keyboard Position Information for 61 Keys	CL	9FH	Initial Touch Data 00H Key OFF 01H Strongest 7FH Weakest
	C#	A0H	
	D	A1H	
	B	DAH	
	C	DBH	

REMARKS

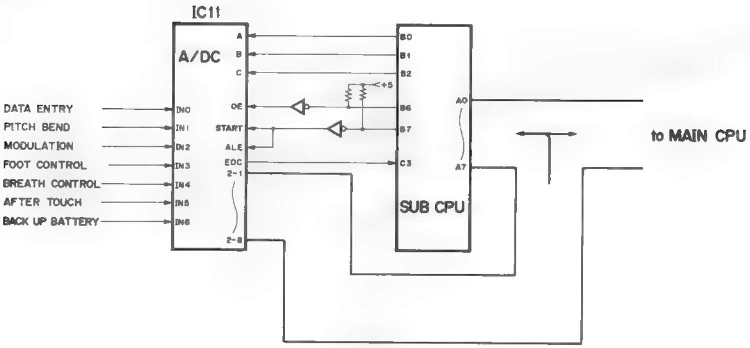
3) A/D Converter Operation

The A/D converter IC converts the analog voltage input from terminals IN_0 ~ IN_7 into 8 bit digital data and outputs that data from the parallel circuit terminals 2–1 ~ 2–8.


Analog Direct Current Input Terminals
1 IN3
2 IN4
3 IN5
4 IN6
5 IN7
Input for A/D START 6 START
Output to display A/D conversion completion 7 EOC
Bit 4 output 8 2 – 5
Input for post A/D conversion digital data output Enable 9 OE
Input for CLOCK that measures speed of A/D conversion speed 10 CLK
+5V 11 Vcc
Reference + voltage input 12 REF (+)
Ground 13 GND
Bit 6 output 14 2 – 7
A/D
Converter
M58990P-1
IN2 28
IN1 27
IN0 26
Detected analog voltage input terminals
A 25
B 24
C 23
Selection data input for A/D conversion input terminals (IN0 ~ IN7)
ALE 22 Input to inform fetch of selection data input to A, B, and C
2 – 1 21 Bit 0 output
2 – 2 20 Bit 1 output
2 – 3 19 Bit 2 output
2 – 4 18 Bit 3 output
2 – 8 17 Bit 7 output
REF (–) 16 Reference – voltage input
2 – 6 15 Bit 5 output

See figures I–7 ~ I–11 (Pages 66, 67 & 68)

A/D Converter Scan Circuit



REMARKS

Remember, A/D conversion is not possible unless the A/D converter is the only unit that has the input of 250KHz. Also, if the EOC of the sub CPU has not returned from the A/D converter, A/D conversion will not be considered to have been completed, and therefore panel switches will not be scanned.

In such cases, the sound from the keyboard can be heard, but after touch is not effective, and panel switch data is not read. Such symptoms are possible when the A/D converter has broken down.

If the 250KHz clock itself breaks down, when the power is turned on the LCD will display the initial message:

* YAMAHA DX7 *
* SYNTHESIZER *

but will not operate after that. Because the status of the internal serial communication interface (SCI) cannot be read and the unit will be inoperative.

The following is an explanation of the flow of operation:

1. With the leading edge of the ALE (Address Latch Enable) of the selection codes for terminals A, B, and C, data is fetched and the terminal for conversion ($\mathrm{IN}_0 \sim \mathrm{IN}_7$) is determined, with the leading edge of the START signal the EOC is reset ("L"), then comes the trailing edge of the start signal and the conversion is started.
   Conversion time changes according to the CLOCK input through the CLK terminal, but in this case it is 250KHz:
   * **Standard — 256 μsec** (100 μsec at 640KHz)
   * **Maximum — 292 μsec** (114 μsec at 640KHz)

The DX7 scans the keyboard during this conversion.

2. This IC outputs EOC (End of Conversion "H" voltage) once A/D conversion has been completed.

3. When the sub CPU senses the EOC, it transmits an OE (Output Enable) and informs terminals 2–1 ~ 2–8 of post conversion data output.

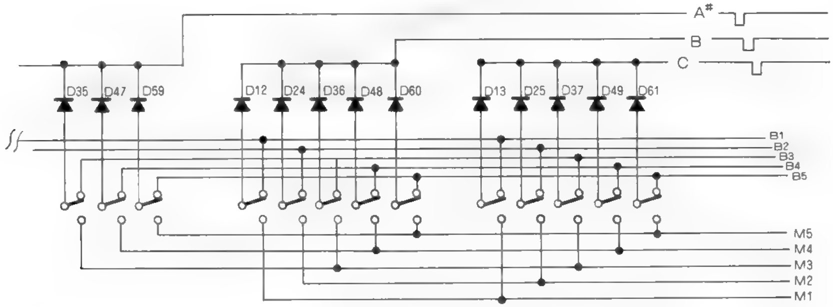
4. When the sub CPU receives the data, it performs an interrupt routine on the main CPU and transmits identifier data to the main CPU.

During Key OFF the above operation is performed for each voltage in 500 μsec intervals, and one cycle is performed during an interval of 4.5 msec.

REMARKS

4) Keyboard Operation

The break (B1 ~ B5) and make (M1 ~ M5) are connected for each octave as shown in this figure:



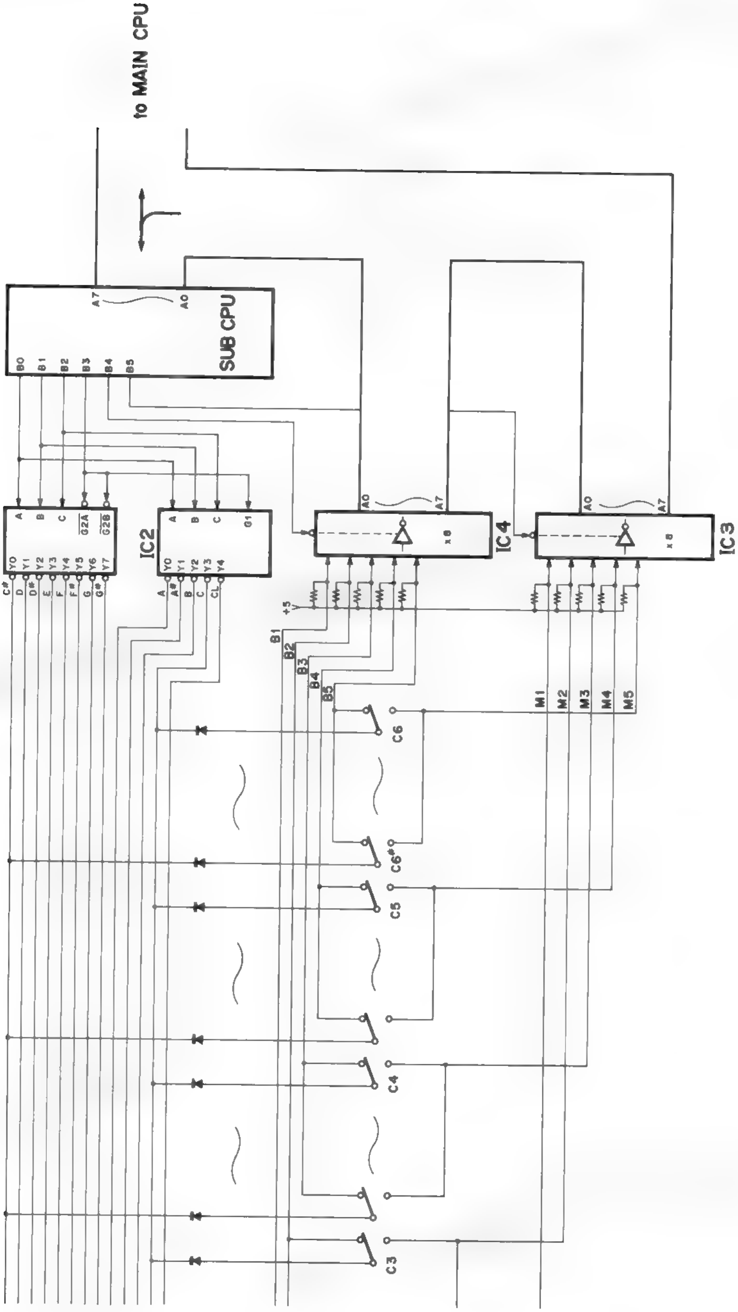
See figures I–9 ~ I–29 (Pages 67 ~ 73)

As you can see, the notes in each octave are scanned simultaneously, including CL this makes a total of 13 notes per octave, and the interval at key OFF is 500 μsec.

(A) The role of Each Contact

- **Break Contact :** To sense the position (note) of each key that is disconnected. To receive "H" voltage from IC_4 of the sub CPU during key OFF. To receive "L" voltage from IC_4 of the sub CPU during key ON.
- **Make Contact :** The time between the breaking of the break contact and the making of the make contact is counted with the timer of the sub CPU and this value becomes initial touch data. During key OFF it receives "L" voltage from IC5 of the sub CPU. During key ON it receives "H" voltage from IC5 of the sub CPU.

Keyboard Scan Circuit



REMARKS

Remember, the followings are three different symptoms that IC1 ~ IC4 can cause:

- ○ When a key is played the display of the LCD changes (because the LCD has been affected by the key code data).
- ○ Some notes in the keyboard cannot be played.
- ○ Panel switch data is not received.

See page 24 for the relationship between the sub and the main CPU.

See figures I–9 ~ I–29 (Pages 67 ~ 73)

The following shows the order of operation:

1. The sub CPU outputs the note selection code from $B_0 \sim B_2$, and the IC selection signal is transmitted from B_3.

B_3	
L level	H level
IC1 Selected	IC2 Selected

	IC1								IC2				
C B A	C#	D	D#	E	F	F#	G	G#	A	A#	B	C	CL
L L L	⊔								⊔				
L L H		⊔								⊔			
L H L			⊔								⊔		
L H H				⊔								⊔	
H L L					⊔								⊔
H L H						⊔							
H H L							⊔						
H H H								⊔					

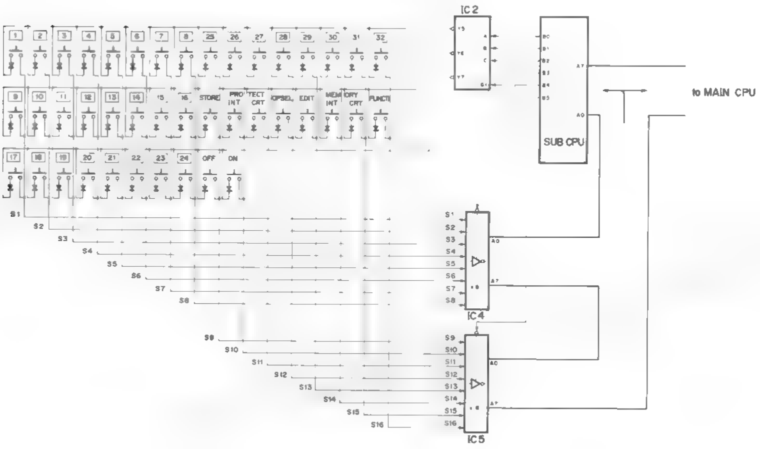
2. B_4 is set at "L", IC4 is enabled and the sub CPU fetches the break data. If the key scanned was OFF, the next key is scanned.

3. If the key scanned was ON, B_5 is set at "L", IC3 is enabled, and the make data is detected; the time this takes is counted and it determines the initial touch value.

5) Panel Switch Operation

When panel switch data is fetched, the switch specification method is the same as for the fetching of the keyboard position information; however, since a maximum 16 switches are connected to one scan terminal (IC_2, SA, SB, SC), $S_1 \sim S_8$ and $S_8 \sim S_{16}$ are fetched two times separately. The panel scan interval is 4.5 msec, and the interval for one cycle is 27 msec.

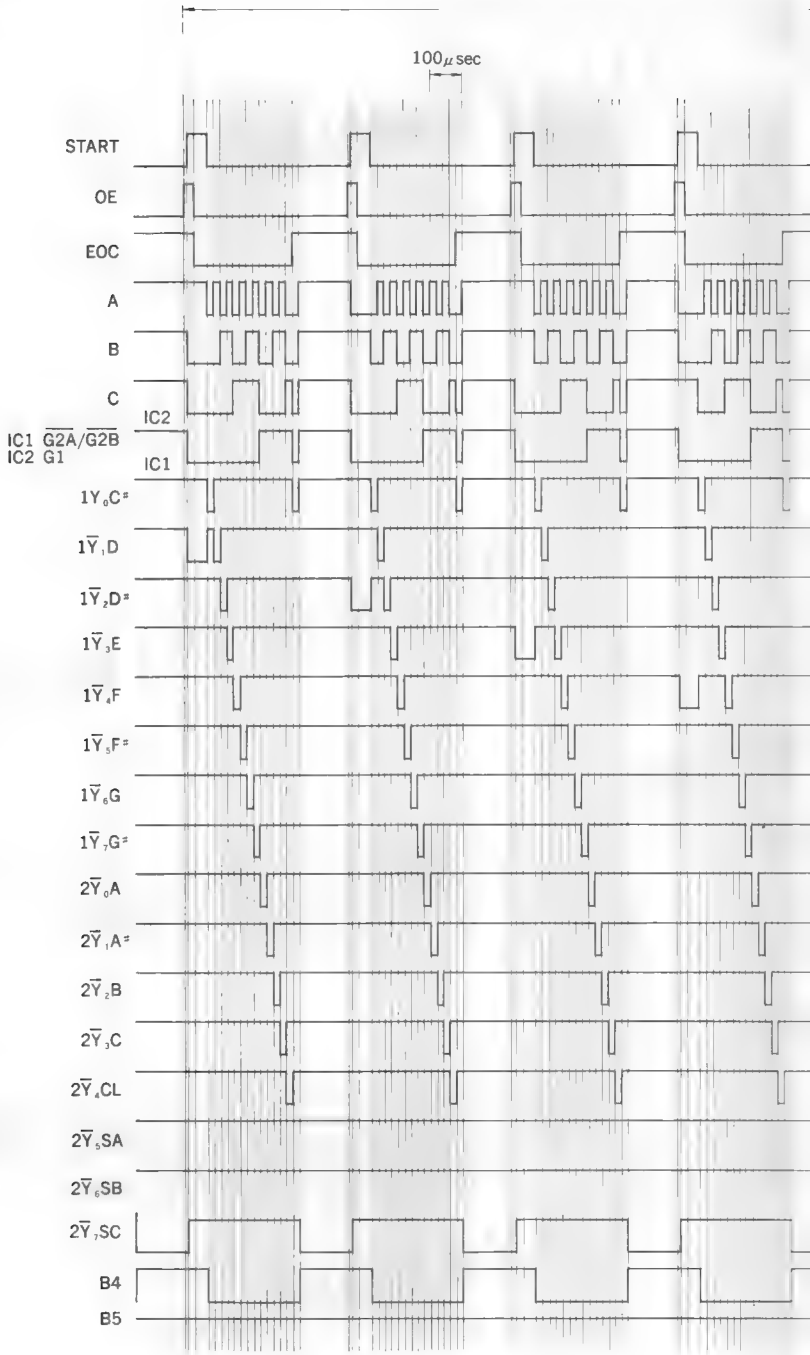
PANEL SW SCAN CIRCUIT



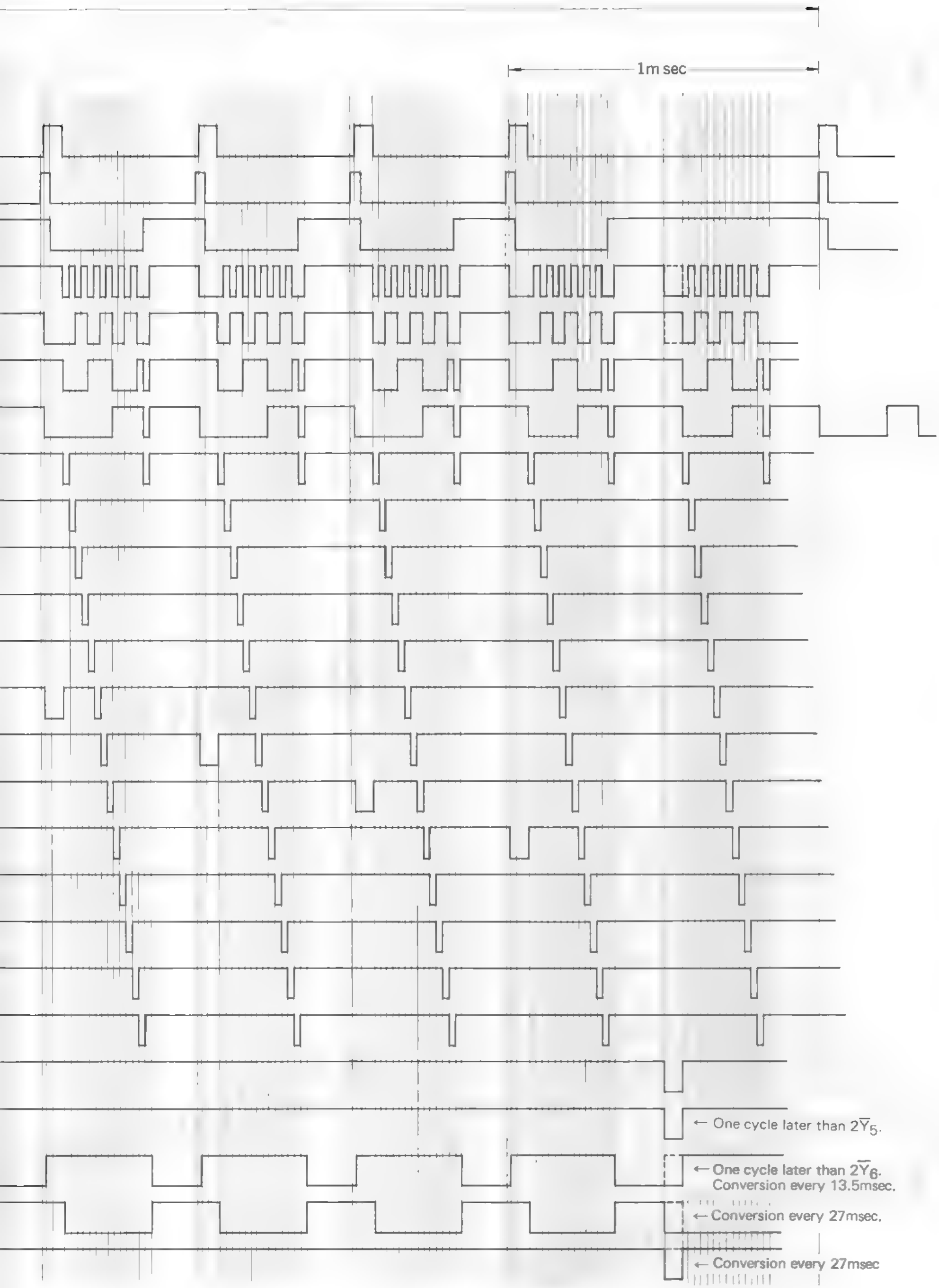
6) Timing Chart for Overall Operation of A/D, Keyboard, and Panel Switches

The following timing chart explains graphically all that has been discussed in this section up to this point. The main points to remember are:

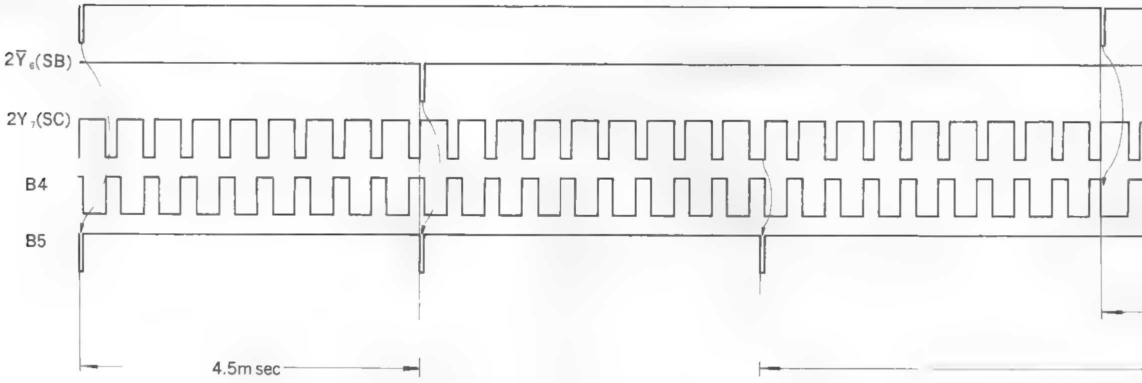
* **Terminals A(B_0), B(B_1), and C(B_2)** are for the binary keyboard, panel switch and A/D specification codes.
* **B_3** "H" means IC2 enabled and "L" means IC1 enabled
* **B_4. B_5** "L" means IC3, IC4 enabled; and keyboard and panel switch data is acquired. Both B_4 and B_5 acquire A/D data while they are set at "H".

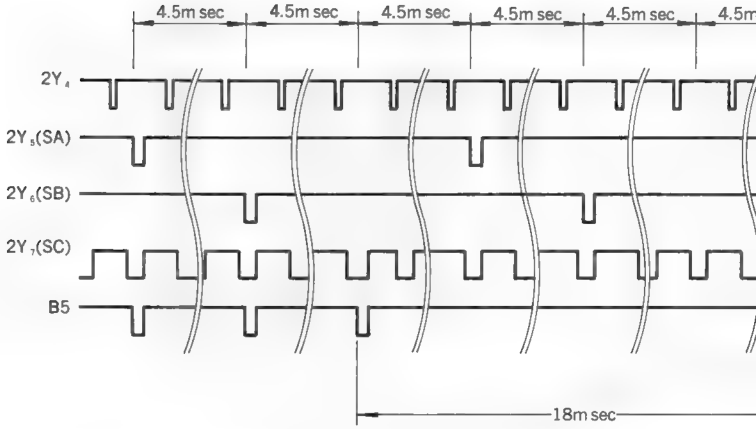
100μsec
START
OE
EOC
A
B
C
IC2
IC1 $\overline{G2A}/\overline{G2B}$
IC2 G1
IC1
$1Y_0C^\sharp$
$1\overline{Y}_1D$
$1\overline{Y}_2D^\sharp$
$1\overline{Y}_3E$
$1\overline{Y}_4F$
$1\overline{Y}_5F^\sharp$
$1\overline{Y}_6G$
$1\overline{Y}_7G^\sharp$
$2\overline{Y}_0A$
$2\overline{Y}_1A^\sharp$
$2\overline{Y}_2B$
$2\overline{Y}_3C$
$2\overline{Y}_4CL$
$2\overline{Y}_5SA$
$2\overline{Y}_6SB$
$2\overline{Y}_7SC$
B4
B5

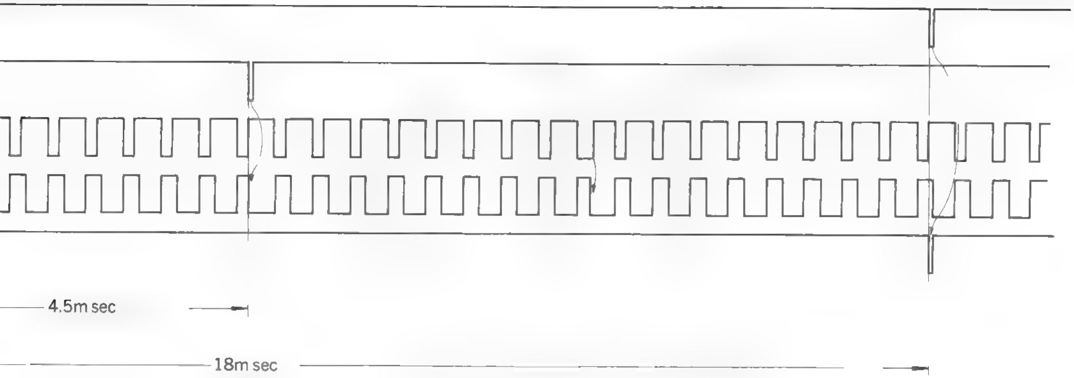
A/D, KEYBOARD AND PANEL SWITCH TIMING CHART

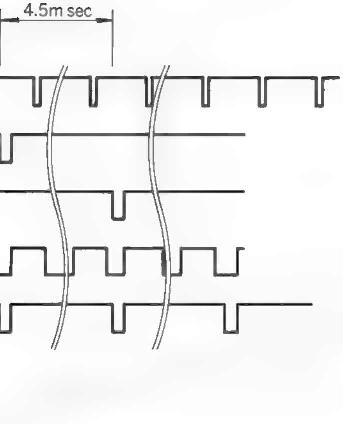

1m sec
← One cycle later than 2$\overline{Y}_5$.
← One cycle later than 2$\overline{Y}_6$.
Conversion every 13.5msec.
← Conversion every 27msec.
← Conversion every 27msec

ENLARGED PANEL SWITCH TIMING CHART

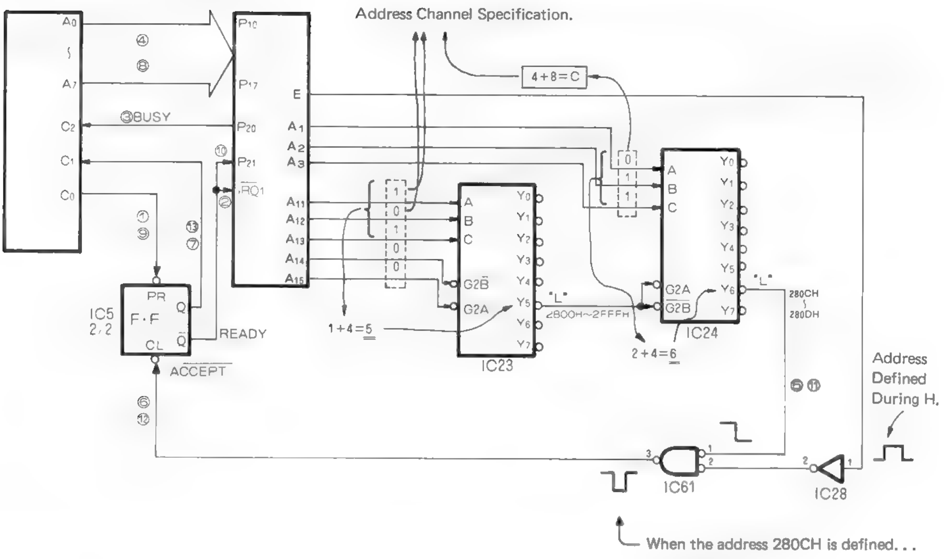





4.5m sec
18m sec


4.5m sec

7) Transmission of Data from the Sub to the Main CPU



The data from the sub to the main CPU is sent in the order numbered in the above figure (1 ~ 13).

1. When there is a change in the data scanned by the sub CPU, it checks its own terminal C_2 (BUSY) and if it is "H" it sets the data at terminals A_0 ~ A_7 and makes C_0 "L". If C_2 is already "L" then the data is stored in the internal RAM (up to 8 columns) before returning to the regular scan routines.

2. Once the C_0 terminal is set at "L", the Flip–Flop (F·F) preset Q terminal becomes "L" and the interrupt routine is performed on the main CPU (IRQ1). Byte 1 of terminal P_{21} is irrelevant.

3. When the main CPU accepts the interrupt routine the data being processed and information that gives the status of the main CPU is evacuated IC21 RAM (in order to be able to continue with the program after the interrupt routine has been completed). Then, with the leading edge of the clock the data input to terminals P_{10} ~ P_{17} is fetched into the main CPU.

4. Once the main CPU has received the first byte of the data, it writes it in address 280CH or 280DH in order to inform the sub CPU. This data can be anything, it has no meaning. At this time, the Y_6 terminal of IC24 becomes "L".

REMARKS

See figures I–4 ~ I–6 (Pages 64 & 65), V–3, 4 (Page 79)

E (Enable): If "H", address is defined, "H" falls to "L" after data has been defined.

5. Pin 1 of IC61 becomes "L", E is "H", pin 3 becomes "L", and the IC5 F·F is cleared.

6. Once cleared, terminal Q of the F·F becomes "L", and the main CPU informs the sub CPU that it has received the first byte of data.

7. After checking to see if sub CPU terminal C_1 has become "L", the second byte of data is output to $A_0 \sim A_7$.

8. The data output to $A_0 \sim A_7$ makes C_0 "L" and inverts the F·F, and Q becomes "L". When Q becomes "L" it may seem that an interrupt routine is being performed, since the data is composed of 2 bytes, it is not an interrupt routine, the object is to set terminal P_{21} at "L". Here is the difference between the first and second byte.

9. The main CPU senses to find out if P_{21} has become "L" or not and determines whether the sub CPU has transmitted the second byte or not and receives the second byte from terminals $P_{10} \sim P_{17}$.

10. ~ 12. The exact same operation is performed as was for the first byte. C_1 of the sub CPU becomes "L" informing that the main CPU has received the second byte of data, and the sub CPU returns to the regular scanning routines.

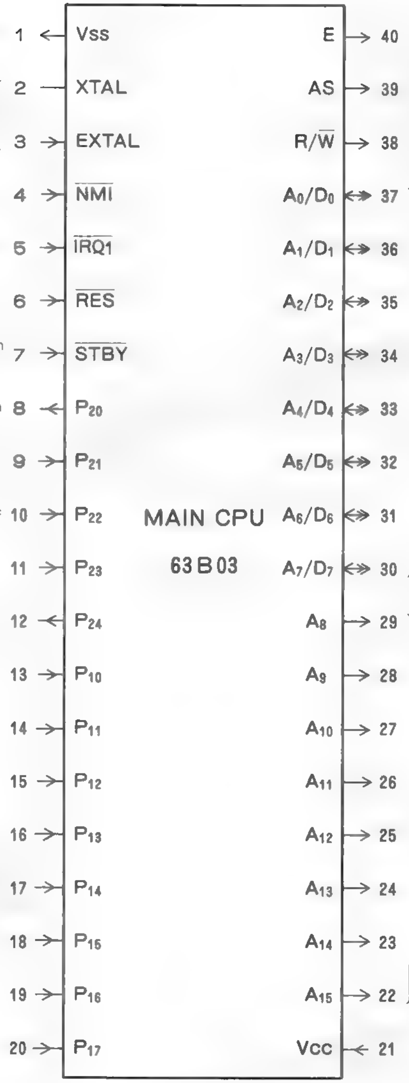
* **Regarding the relationship between the sub and main CPU . . . . .**

Since the main and sub CPU handshake in order to transmit and receive data, the main CPU merely operates along with the data codes sent from the sub CPU. So it is not strange that the LCD display should change when the keyboard is played, it shows that the main CPU is working normally. If data should change while in the circuits of the main CPU it will not operate properly from the start. Also, it is very unlikely that something is wrong with the sub CPU if it is sending the proper data (sending data properly); rather it is very possible that the decoder or buffer is faulty.

Ⅲ THE FM TONE GENERATOR

1) The Microcomputer

Before understanding how the circuitry around the main CPU operates, it is necessary to understand the functions of the main CPU's terminals.



GND

EXTAL terminals are used when a CLOCK is supplied from outside.

Interrupt input, unused

Input for data that the sub CPU wants to interrupt main CPU for

Rest terminal for internal initialization

Terminal for low power consumption mode, unused

Terminal to inform status of main CPU, busy at "L"

Input for acknowledging if sub CPU is sending data to main CPU ("L")

Clock terminal to specify speed of transmission of MIDI data

MIDI input

MIDI output

Inputs for data from sub CPU

Output terminal to inform of address and data definition; 1/4 the frequency of the CLOCK is output. H means address, H → L means data definition.

Output terminal to inform of transmission of lower 8 bits of address; definition when H → L.

Read/write output control terminal, "H" = read (receive), "L" = write (transmit).

Shared output for data bus and lower 8bits of address. After output of address, data is input/output.

Output for upper 8 bits of address

Power +5V

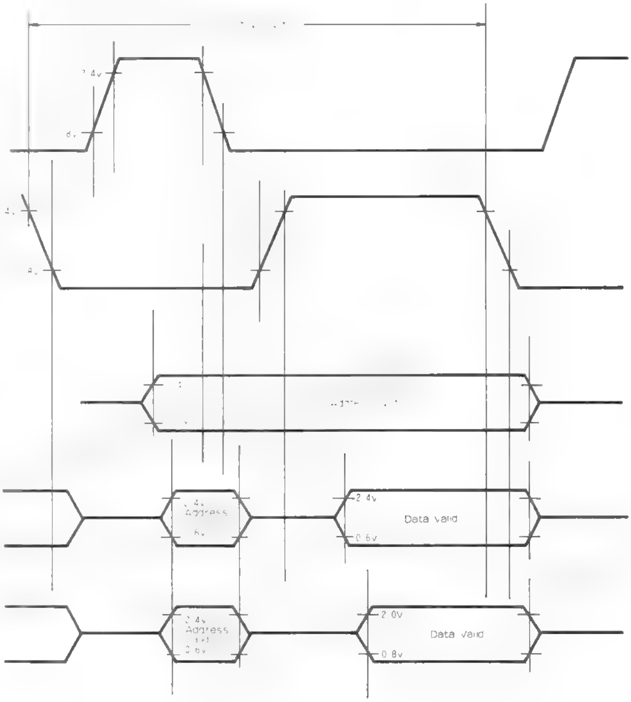
REMARKS

See figures III–1 ~ III–7 (Pages 75 & 76)

- ○ Unless 250KHz is input into the CPU, operation will stop after the output of the initial message.
- ○ If IC7, the CPU mode setting IC, has broken down, there will be no initial message nor operation.
- ○ If the MIDI data at terminal P_{23} is always "L", then the operation of the CPU will be stopped.

In such cases the CPU itself does not really stop operation, the DX7 simply stops functioning correctly.

Since the CPU data bus and lower 8 bits of the address bus are used for various different purposes, there is an AS (Address Strobe) terminal to distinguish if signals are meant for $AD_0 \sim AD_7$ or if they are the lower 8 bit address. It is essential to understand this bus timing in order to understand the operations of the units peripheral to the CPU.



The timing of the above diagram is explained in detailed as follows:

- **AS [Same as ALE (Address Latch Enable]**

  During both read and write operations the lower 8 bits of the address must be latched at the trailing edge of the pulse. This is because it is possible that the lower 8 bit address may not be valid at the leading edge or while AS is high.

- **E (Enable)**

  This CLOCK is 1/4 of the CLOCK that is input at the EXTAL terminal, and it determines the timing for operations outside the circuit. R/W is also defined before the leading edge of E. During writing operations data is valid at the trailing edge of E.

REMARKS

If the master clock is not input, the CPU will not operate at all.

During reading operations the data must be valid before the trailing edge of E.

- **R/W (Read/Write)**

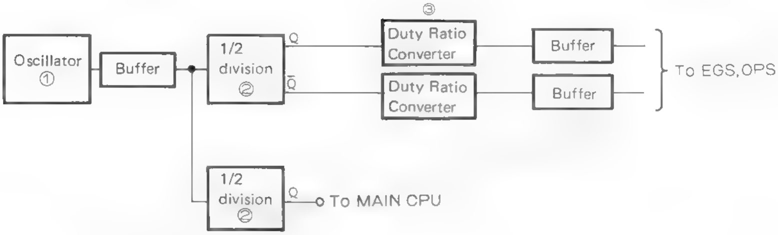
  Read operations (commands, data, etc,; operations which fetch into the CPU) output high voltage.

  Write operations (post arithmetic operation data, etc.; operations which output from the CPU) output low voltage.

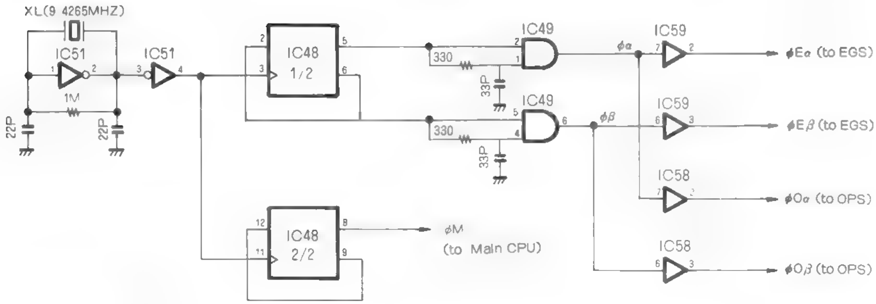
  These control signals are valid before the leading edge of E.

(1) Master Clock 1

This circuit creates the clock for the OPS, EGS, and main CPU and the DX7 uses this clock to valid timing for operations. The circuit for the clock is shown in the figure.

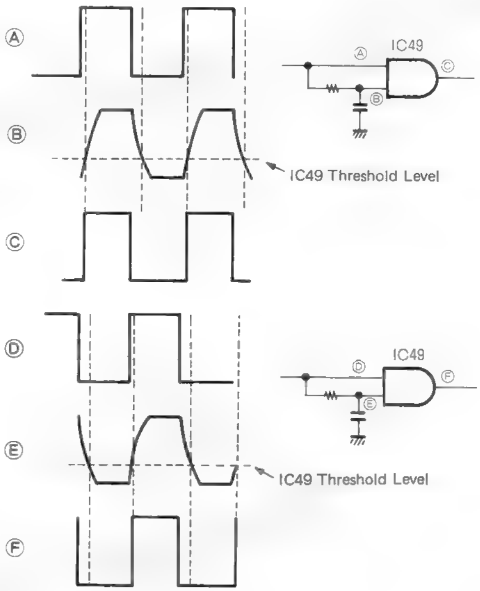


Master Clock Oscillator **Two Phase Clock Driver**



Master oscillation is 9.4265MHz, but D flip flop becomes T flip flop, and thus in 1/2 division, the resulting system clock is thus: 4.71325 MHz (T=1/f≒212.17 nsec). The system clock for EGS and OPS is varied by its duty ratio.

As shown in the figure, the IC49 threshold level and integration by the C and R waveforms are used for duty ratio variation.

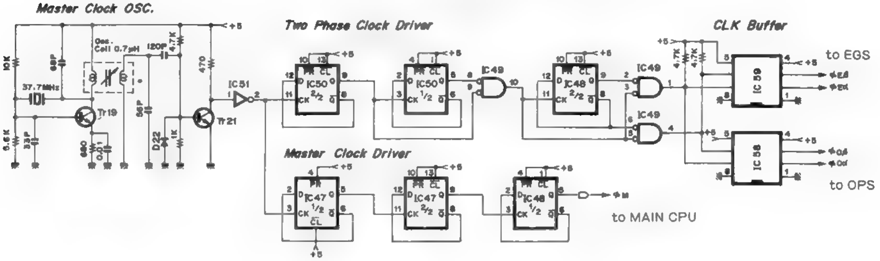


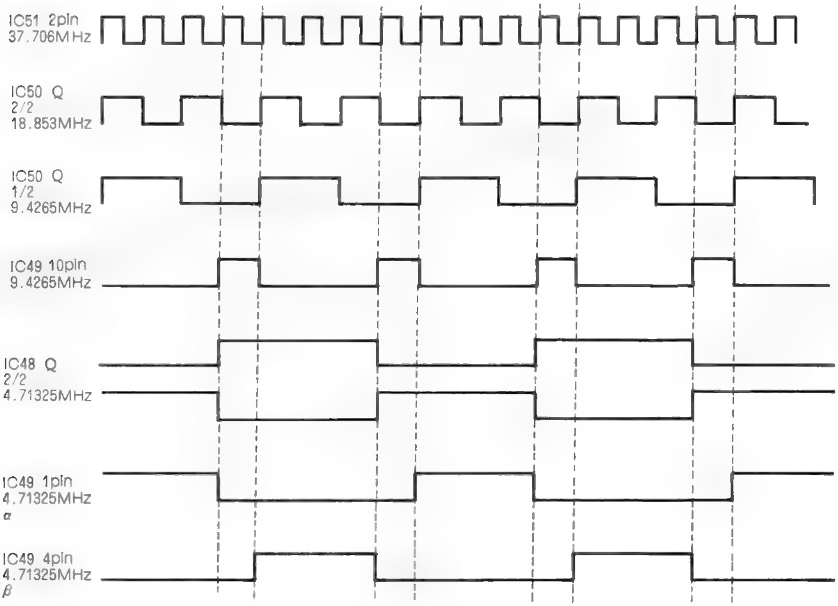
REMARKS

See figures II–1 ~ II–4

Master Clock 2

The diagram below is the circuit for units with serial numbers from #1001 to 2478, and 2661 to 2842. The timing chart leads up to a 2 phase clock, follow the logic to see.





REMARKS

See figures IV–3, IV–4 (Page 77)

(2) ROM Read Operations

The ROM chip used for DX7 has memory capacity of 64Kbit (8Kbyte). The space in the CPU for addresses is a maximum of 64Kbytes; however, it is necessary to know where the address has been stored within that space. The address decoder performs address assignment of ROMs, and the assignment is performed by the TTL gate.

This circuit will now be explained.

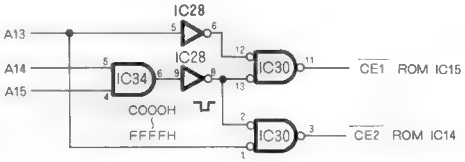
Since the ROM is 8Kbytes, and if there are 13 address lines ($A_0 \sim A_{12}$), the entire memory area of the ROM can be assigned.

Address line used in ROM

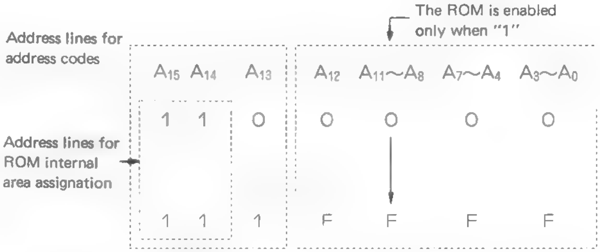
$$2^{13} = 8192\text{byte}$$

$$8192 \div 1024 = 8\text{Kbyte}$$

Therefore, if an address code is performed with the remaining three lines (A_{13}, A_{14}, A_{15}) assignment can be performed randomly in 8Kbyte blocks within the 64Kbyte memory space.



And operation has been performed on address lines A_{14} and A_{15}, so a high logic level is output only when both of them are high together. This means the following:


The ROM is enabled only when "1"
Address lines for address codes
A15 A14
A13
A12
A11~A8
A7~A4
A3~A0
1 1
O
O
O
O
O
Address lines for ROM internal area assignation
1 1
1
F
F
F
F

In other words, all terminals other than A_{14} and A_{15} can be either all 1 or all 0.

So:

When $A_{13} \sim A_0$ are all 0 — COOH

When $A_{13} \sim A_0$ are all 1 — FFFFH

REMARKS

A_{15} A_{14} A_{13} A_{12}	A_{15} A_{14} A_{13} A_{12}	
1 1 0 0	8+4	C
1 1 0 1	8+4+1	D
1 1 1 0	8+4+2	E
1 1 1 1	8+4+2+1	F

A high logic level is produced on the Pin 6 of the address bus of the IC$_{34}$ only when addresses from COOOH to FFFFH are output. Therefore, the output for the IC30 (AND gate) which comes afterward is low only in such cases. And the IC30 pin 1 is connected to A_{13} directly, while pin 12 is A_{13} inverted, so it becomes the following. IC30 pin 3 is low only when A_{13} is low, so:

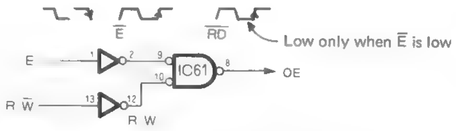
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}~A_8	A_7~A_4	A_3~A_0
1	1	0	0	0	0	0
			↓	↓	↓	↓
1	1	0	1	F	F	F

Since this is COOOH ~ DFFFH (8Kbytes) this address space of ROM IC14 becomes assigned. Because ROM IC15 is selected when A_{13} is High,

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}~A_8	A_7~A_4	A_3~A_0
1	1	1	0	0	0	0
			↓	↓	↓	↓
1	1	1	1	F	F	F

EOOOH ~ FFFFH (8Kbytes) address space becomes assigned. Address data from within these limits is selected and output to the address bus, and the ROM is selected, however, the ROM has only been made open to the selection operation, nothing has been output on the data bus. The $\overline{OE}$ (Output Enable) terminal is provided for output to the data bus, when this terminal is low the data of the area defined by the address is output.

In order to set terminal $\overline{OE}$ at low, the E and $R/\overline{W}$ signals from the CPU are inverted, and the result is transmitted as an $\overline{RD}$ signal using the AND gate.



Since $\overline{OE}$ is negative logic, E and $R/\overline{W}$ are inverted, and only when both $\overline{E}$ and $\overline{R}$ are low at the input of the AND gate does the AND operation allow the output of the memory contents. In the CPU, when E falls from H to L (the same instant that terminal OE becomes high), the data on the data bus is fetched. At this time, between the

REMARKS

128Kbytes ÷ 8 = 16Kbytes

2^{14} = 16384bytes

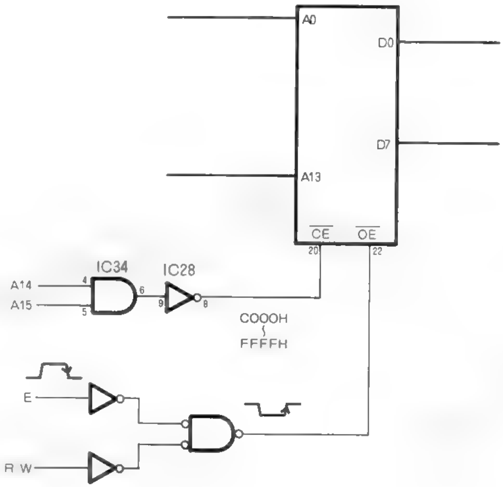
16384 ÷ 1204 = 16Kbytes

ROM, RAM, etc., and the data bus of the CPU the bi-directional bus buffer IC32 is inserted.

Terminal $\overline{G}$ becomes low, and at that time, for the DIR (direction) terminal —

H = output from CPU
L = input to CPU

Next is the ROM circuit for units with serial numbers from #2661 ~. Units with serial numbers under #2661 use two 64Kbit EPROMs, but from serial number #2661 on a 128Kbit mask ROM is used. One IC suffices.



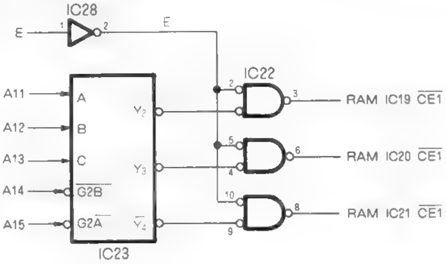
REMARKS

See figures IV–1, IV–2, IV–5 (Pages 77, 78)

See figures IV–1, IV–2, IV–5 (Pages 77, 78)

3) RAM Read/Write Operations

For the RAM, three chips of 16Kbit (2Kbyte) CMOS IC are used. It is backed up with a battery so that even if the power is turned off the memorized data are preserved. To understand how the RAM address space is assigned, it is easiest to think of it as a ROM.



The IC23 is a 3 to 8 Demultiplexer, and is used as the address decoder. Only when $\overline{G_2A}$ and $\overline{G_2B}$ are low at the same time one of terminals $\overline{Y_0}$ ~ $\overline{Y_7}$ is selected according to the codes input at A, B, or C. The terminal selected becomes low.

RAM IC19 : $\overline{Y_2}$ = LOW
RAM IC20 : $\overline{Y_3}$ = LOW
RAM IC21 : $\overline{Y_4}$ = LOW

A	B	C	$\overline{G_2A}$	$\overline{G_2B}$	G_1	Output
0	0	0	0	0	1	$\overline{Y_0}$
1	0	0	0	0	1	$\overline{Y_1}$
0	1	0	0	0	1	$\overline{Y_2}$
1	1	0	0	0	1	$\overline{Y_3}$
0	0	1	0	0	1	$\overline{Y_4}$
1	0	1	0	0	1	$\overline{Y_5}$
0	1	1	0	0	1	$\overline{Y_6}$
1	1	1	0	0	1	$\overline{Y_7}$

At this time the RAM are operational as shown in the following table:

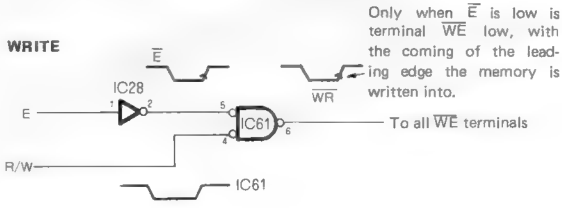
	$\overline{G_2A}$	$\overline{G_2B}$	C	B	A	RAM Address bus										
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
$\overline{Y_2}$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
							∫			∫				∫		
	1				0	1	1	1	1	1	1	1	1	1	1	1
$\overline{Y_3}$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
							∫			∫				∫		
	1				1	1	1	1	1	1	1	1	1	1	1	1
$\overline{Y_4}$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
							∫			∫				∫		
	2				0	1	1	1	1	1	1	1	1	1	1	1

In the above table –

$\overline{Y_2}$: 1000H ~ 17FFH (RAM IC19)
$\overline{Y_3}$: 1800H ~ 1FFFH (RAM IC20)
$\overline{Y_4}$: 2000H ~ 27FFH (RAM IC21)

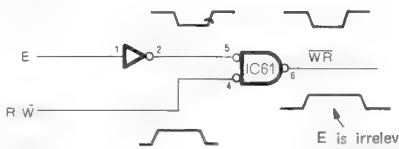
The inverted E with $\overline{Y_2}$, $\overline{Y_3}$ and/or $\overline{Y_4}$ are applied to AND to select RAM. The read/write terminal of this RAM is used for both reading and writing; when it is low, writing operations are performed, when it is high, reading operations are performed.

WRITE



Only when $\overline{E}$ is low is terminal $\overline{WE}$ low, with the coming of the leading edge the memory is written into.

READ



E is irrelevant, only when H on R/$\overline{W}$ dominates

4) The Role of the RAM

The three RAM are used in the following ways:

RAM IC19 }
RAM IC20 } Internal memory for voices; they hold the parameter data and the display data for 1 ~ 32 voices.

They are basically used for reading operations and to change the data they contain with writing operations and it is necessary to use store and load operations.

RAM IC21 This RAM can be called a work RAM, it aids when there is not enough working space (internal registers) in the CPU. It is both read and written in regularly, with the two RAM, they are an extremely important part of the system.

- o **Voice parameter Data Memory during EDIT**
- o **Memory of Status of CPU when Interrupt Routines received from sub CPU.**
- o **Memory of Status of previous to Power OFF.**
- o **Data Memory during FUNCTION mode.**
- o **Memory Protect for Identifier Data.**

These are only some of the functions that this important unit performs.

5) About P_{22}, P_{23}, and P_{24}

The main CPU of the DX7 is equipped with an SCI (Serial Communication Interface) and so it is able to send and receive data serially and is not synchronized with the exterior. This function is used for the MIDI. When the SCI receives MIDI data and interrupts the main CPU from within to call the MIDI routine (internal terminal IRQ2 has a lower priority than the external interrupt terminal IRQ1).

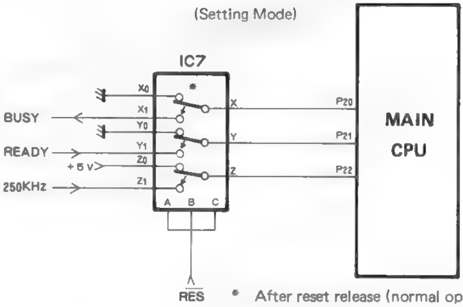
P_{22} is the input terminal for the clock that is used to determine the speed of the MIDI data transmission, the speed of the transmission (BAUD rate) is 1/8 of the frequency.

250KHz ÷ 8 = 31,250 (Baud)

This speed of 31,250 Baud means that 31,250 bits are sent in the space of 1 second.

P_{23} is the receiving terminal for MIDI data
P_{24} is the transmission terminal for MIDI data

6) Main CPU Mode Setting Circuit



* After reset release (normal operaiton), SW selects as shown by arrows.

This CPU has 2 modes, selection of either mode depends on what kind of voltage is input to terminals P_{20}, P_{21}, and P_{22} when the low voltage of reset becomes high. The DX7 decides the mode according to the combination of terminals X_0, Y_0, and Z_0.

In this case the mode is the selection of **whether or not to use the lower 8bits for both data and address,** and depending on the mode the ports below are used:

- **Expanded Multiplex Mode**
  P_{30} ~ P_{37} Data Address Lower 8bits used
  P_{10} ~ P_{17} Used as ports

- **Expanded Nonmultiplex Mode**

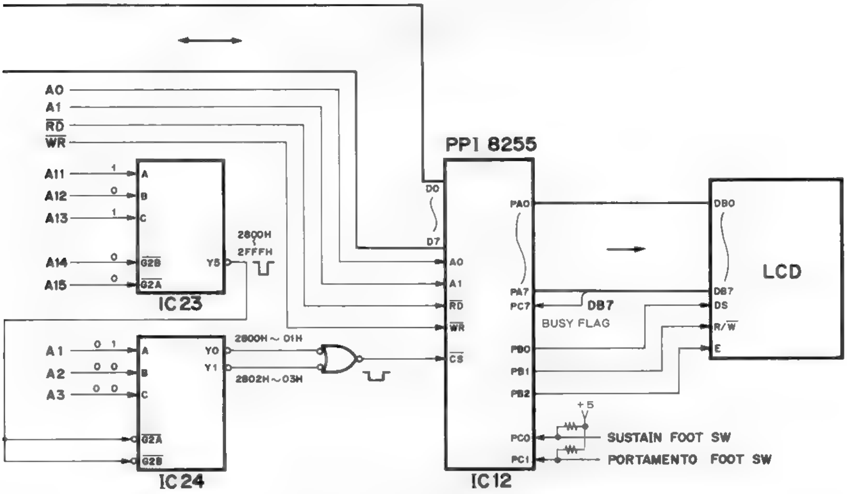
  $P_{30} \sim P_{37}$ Used as data line

* $P_{10} \sim P_{17}$ Used for lower 8 bits of address

Now, the **DX7 also uses the expanded multiples mode** and $P_{10} \sim P_{17}$ to recieve data from the sub CPU.

2) LCD Interface (including Sustain and Portamento)

LCD
Sustain Interface Circuit
Portamento

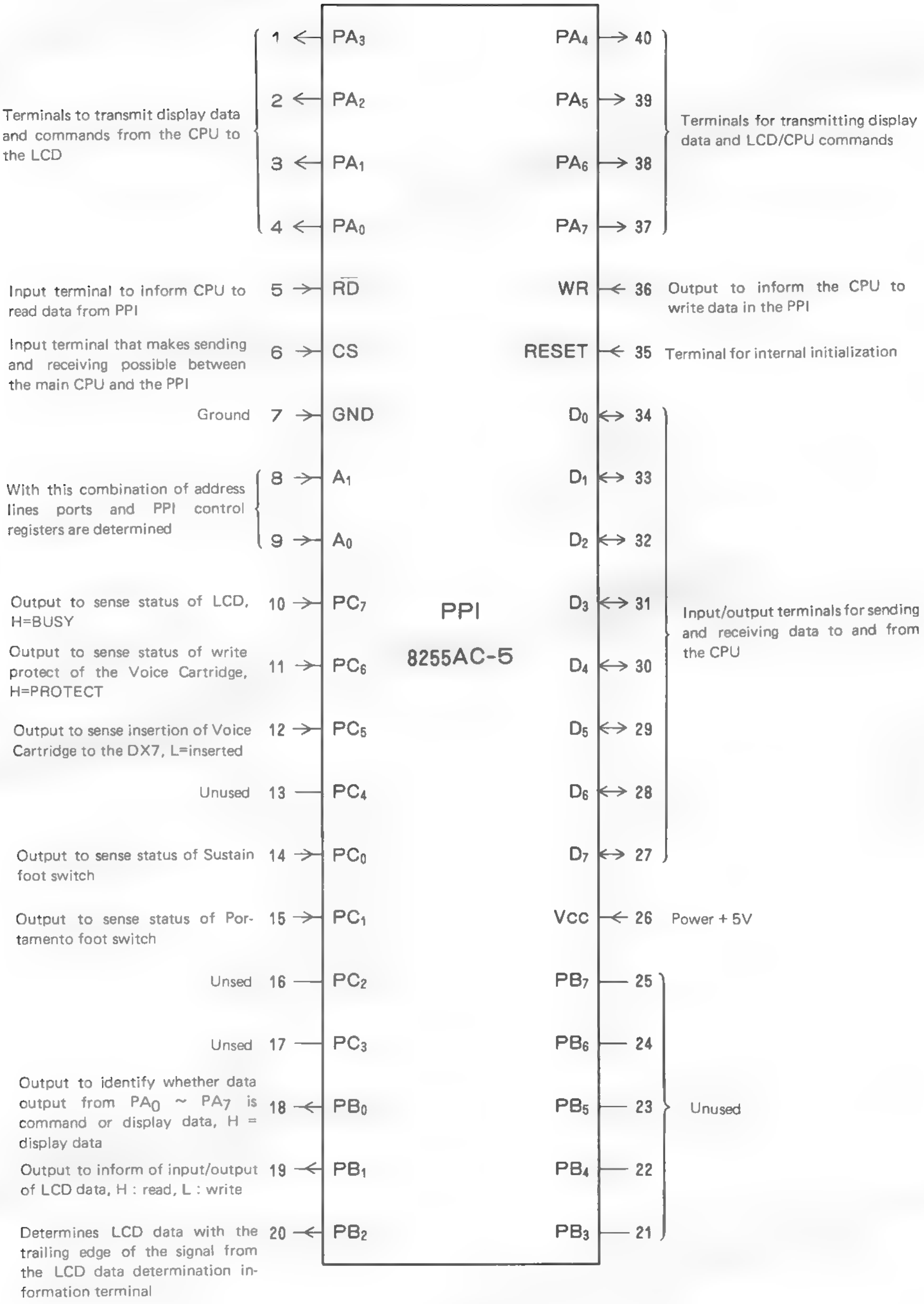


1) PPI

See figures V–1, V–2 (Page 79)

PPI stands for Programmable Peripheral Interface, and it is the interface for the LCD. With a program, each port (PA_0 ~ PA_7, PB_0 ~ PB_7) can be set as input or output, and they are equipped with latches and buffers.

PA_0 ~ PA_7 : output
PB_0 ~ PB_7 : output
PC_0 ~ PB_7 : input port C data is not latched)



Terminals to transmit display data and commands from the CPU to the LCD

Input terminal to inform CPU to read data from PPI

Input terminal that makes sending and receiving possible between the main CPU and the PPI

Ground

With this combination of address lines ports and PPI control registers are determined

Output to sense status of LCD, H=BUSY

Output to sense status of write protect of the Voice Cartridge, H=PROTECT

Output to sense insertion of Voice Cartridge to the DX7, L=inserted

Unused

Output to sense status of Sustain foot switch

Output to sense status of Portamento foot switch

Unsed

Unsed

Output to identify whether data output from PA_0 ~ PA_7 is command or display data, H = display data

Output to inform of input/output of LCD data, H : read, L : write

Determines LCD data with the trailing edge of the signal from the LCD data determination information terminal

Terminals for transmitting display data and LCD/CPU commands

Output to inform the CPU to write data in the PPI

Terminal for internal initialization

Input/output terminals for sending and receiving data to and from the CPU

Power + 5V

Unused

REMARKS

If for some reason the LCD and DB7 lines become stuck at H, when the power is turned on, the initial message will not be displayed and the unit will not operate.

This is because DB7 is using a BUSY FLAG. As long as it is H, the CPU will think the LCD is BUSY, and wait for an L.

PPI addresses are:

2800H ~ 2803H

This uses the same method as for the RAM. The PPI is enabled by IC24 in the addresses 2800H ~ 2FFFH. Addresses A_1, A_2, and A_3 are decoded, so:

When $\overline{Y_0}$ is 2800H ~ 01H : low
When $\overline{Y_1}$ is 2802H ~ 03H : low

And the Y_0 and Y_1 outputs are sent via an OR gate, so when PPI is addressed at 2800H ~ 2803H is it operational.

The following is a list of the ports, addresses and their related functions:

2800H-Port A-Outputs data on data bus to port A at $\overline{WR}$

2801H-Port B-Outputs data on data bus to port B at $\overline{WR}$

2802H-Port C-Outputs data on data bus to port C at RD

2803H-PPI Program (PPI Mode selection, Port selection for Input/Output)

2) The LCD

* DS (RS) terminals determine whether the data input at terminals DB_0 ~ DB_7 is a command or display data.
* When input at the R/$\overline{W}$ terminal is R(Read=high), the RAM data can be read, or the LCD busy flag can be sensed. In the case of the DX, only the busy flag is read. At $\overline{W}$(Write=low) commands and display data is written in the LCD.
* The E(Enable) terminal defines input data and commands on the bus with the trailing edge of H → L.

3) C Port Usage

* PC_7 is connected to DB_7 of the LCD, with DS at low and R/$\overline{W}$ at high, DB_7 will show the operational status of the LCD, High=BUSY. When it is busy the CPU may not transmit data or commands to the LCD.
* PC_0 and PC_1 are connected to the sustain and portamento foot switches and are usually pulled up at +5V. When their switches are turned on, the terminals become low. The CPU detects whether the switches have been turned on or not. This is the lowest priority port of the CPU.

REMARKS

To write data into the EPROM 21 to 25V DC is necessary.

See figure V–7 (Page 80)

3) Cartridge Interface Operation

There are 2 types of cartridges:

- **64Kbit (8Kbyte) EPROM x 1**
- **16Kbit (2Kbyte) EEPROM x 2**

The EPROM is the same type that is used within the DX7, the EEPROM (Electrically Erasable PROM) is sold as a RAM backup. Not only is the EEPROM electrically erasable, but it does not require the high voltage that EPROMs require for writing operations, therefore eliminating the necessity for the special equipment that is needed to write in the EPROM.

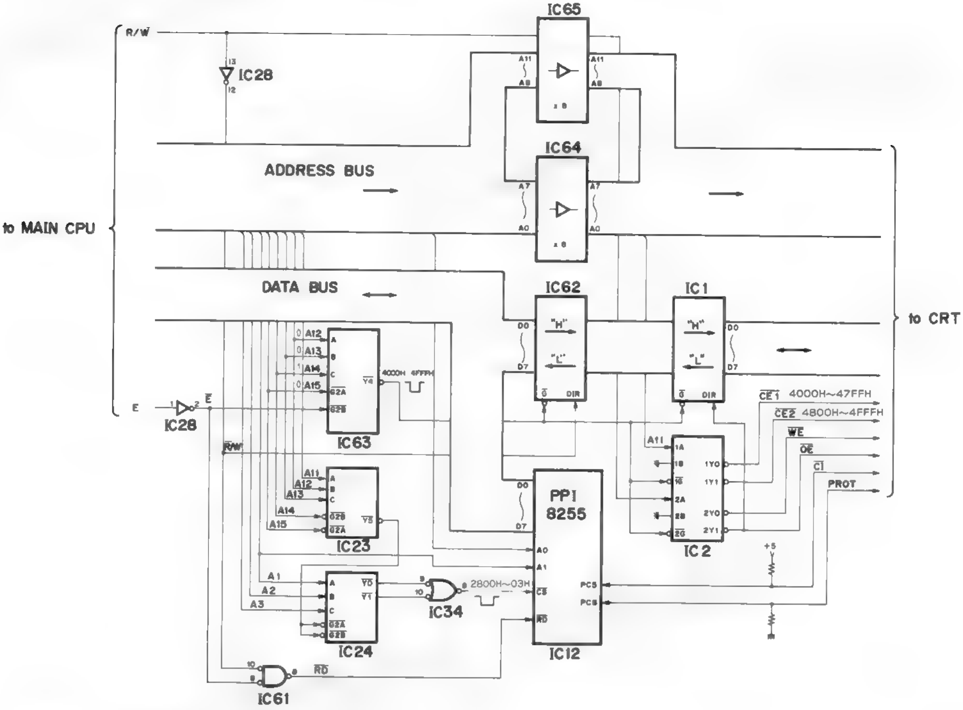
Of course, since these units are ROM, there is no need for a backup battery. The EEPROM can be used as a sort of protected RAM, but it is different from a conventional RAM in the following ways:

10msec is needed for a one byte write cycle.

The maker guarantees the unit for 10,000 writing operations.

Therefore in order to write 4Kbytes of data –
10msec x 4,096 = 40sec
However, this figure does not include the time it takes to execute the writing program. Since the unit is used as external voice data memory, there is little worry about the limitations on writing.

Cartridge Interface Circuit



1) Voice Cartridge Status Sensing

The main CPU goes through terminals PC_5 and PC_6 of the PPI to sense the status of the Voice Cartridge before any commands or data are sent.

- **PC_5 is $\overline{CI}$ (Cartridge Input)**
- **PC_6 is PROT (PROTECT)**
- **CI senses whether the cartridge is inserted or not.**
  When it is, CI becomes low.
- **PROT senses the status (ON-OFF) of the switch of the memory protect of the RAM Pack.**
  When the switch is high it is ON.

2) CRT Read/Write Operation

The memory area of the Voice Cartridge is assigned the following addresses by the decoder (IC63):
4000H ~ 4FFFH (4Kbytes)
Terminal Y_4 becomes low when these addresses are valid, operation

of the bi-directional buffers (IC62, IC1) and the decoder (IC2) are enabled. The CPU and Voice Cartridge data bus and control lines are connected, and data can be sent back and forth.

The address bus is connected to the Voice Cartridge through the buffers IC64 and 65.

*** READ (Voice Cartridge ⟶ CPU)**

When the DIR (direction) terminal of the bi-directional buffer IC62 becomes low, data is sent from the Voice Cartridge to the CPU. Since the DIR terminal of the bi-directional buffer IC1 is connected to terminal $2Y_1$ of the decoder IC2, IC1 changes to the read direction by making terminal $2Y_1$ low. This happens when terminal 2A of IC2 is high, and this terminal is connected to the R/W terminal of the CPU. It becomes high during read operations.

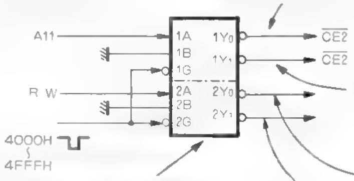
Terminal 1A of IC2 is connected to address line A11, and terminal 1B is grounded. When –

$1Y_0$–4000H ~ 47FFH (2Kbytes)

$1Y_1$–4800H ~ 4FFFH (2Kbytes)

the terminal is low.

When address line A11 (terminal 1A) is low, meaning 0000H ~ 07FFH are selected, this output should become low. However, addresses from 4000H on of this IC are enabled; so, actually only when 4000H ~ 47FFH are selected does $1Y_0$ become low.



When address line A11 (terminal 1A) is high, meaning 0800H ~ 0FFFH are selected, this output should become low. However, addresses from 4000H on of this IC are enabled; so, actually only 4000H ~ 47FFH and $1Y_1$ become low.

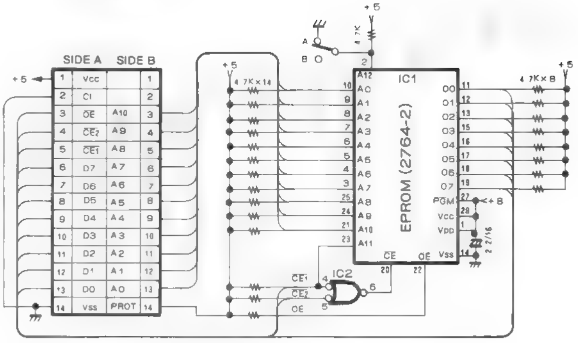
This IC will not operate for addresses 4000H ~ 4FFFH

Becomes low when terminal 2A becomes low (write)

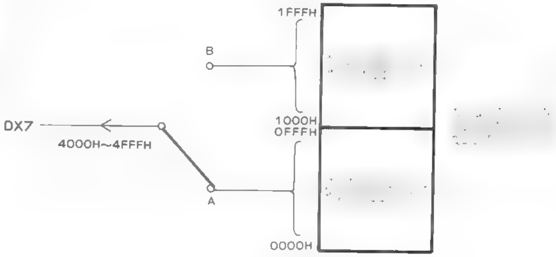
Becomes low when terminal 2A becomes high (read)

3) Voice Cartridge

*** ROM cartridge**

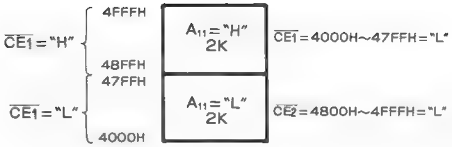


The ROM cartridge has a memory capacity of 8Kbytes for storing 64 voices, but the cartridge is provided with a memory area of only 4Kbytes (addresses 4000H ~ 4FFFH) for 32voices. Selection between the two groups of 4Kbyte memory areas is made by mechanically switching the uppermost address line A_{12} of the ROM cartridge to L or H. This allows the entire 64 voices stored in the ROM cartridge to be available to the user.



The address lines connected to the cartridge are A_0 ~ A_{10}. This alone allows only 2Kbyte memory to be addressed since 2^{11} equals 2,048. Two chip enable lines, $\overline{\text{CE1}}$ and $\overline{\text{CE2}}$ are provided for selecting two sets of 2Kbyte memories, thus enabling a total of 4Kbyte memories to be accessed by the user.

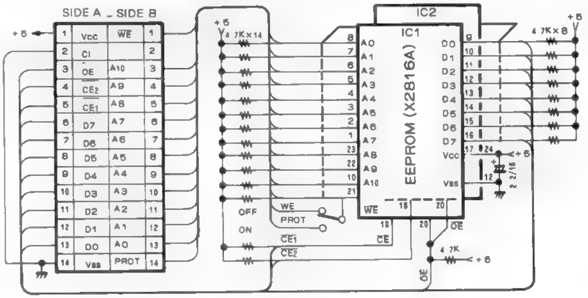
The chip enable line ($\overline{\text{CE}}$) is the result of an OR operation between the chip enable lines $\overline{\text{CE1}}$ and $\overline{\text{CE2}}$ and allows the memory area : 4000H ~ 4FFFH to be accessed. The address line A_{11} is connected to the chip enable line $\overline{\text{CE1}}$, which is based on the same principle as the handling of the address line A_{12} mentioned earlier. When the chip enable line


4FFFH
$\overline{\text{CE1}}$="H"
48FFH
47FFH
$\overline{\text{CE1}}$="L"
4000H
A_{11}="H"
2K
A_{11}="L"
2K
$\overline{\text{CE1}}$=4000H~47FFH="L"
$\overline{\text{CE2}}$=4800H~4FFFH="L"

$\overline{\text{CE1}}$ is set to L, the address line A_{11} is also set to L and the memories 4000H ~ 47FFH are selected. The ROM itself is enabled by the chip enable line $\overline{\text{CE1}}$(L). When the chip enable line $\overline{\text{CE1}}$ is set to H, the are selected. When $\overline{\text{CE1}}$ is H, $\overline{\text{CE2}}$ is L (naturally, momory locations 4000H ~ 4FFFH must be selected), thus allowing the ROM to be enabled.

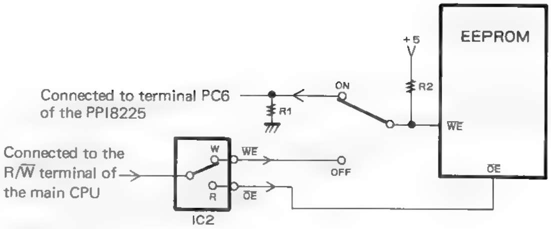
The input line $\overline{\text{CI}}$ is connected to the ground within the cartridge. Hence, terminal PC5 of the PPI18255 becomes L immediately after the cartridge is connected into DX7

* RAM cartridge



Two EEPROM (2Kbytes each) are incorporated into the ROM cartridge and the address and chip enable lines are provided in normal ways. Thus, it is not necessary to comment on them at all.

Next, let's talk about the protect switch provided with the RAM cartridge.


EEPROM
+5
R2
ON
Connected to terminal PC6 of the PPI8225
R1
$\overline{\text{WE}}$
Connected to the R/$\overline{\text{W}}$ terminal of the main CPU
W
WE
OFF
$\overline{\text{OE}}$
R
$\overline{\text{OE}}$
IC2

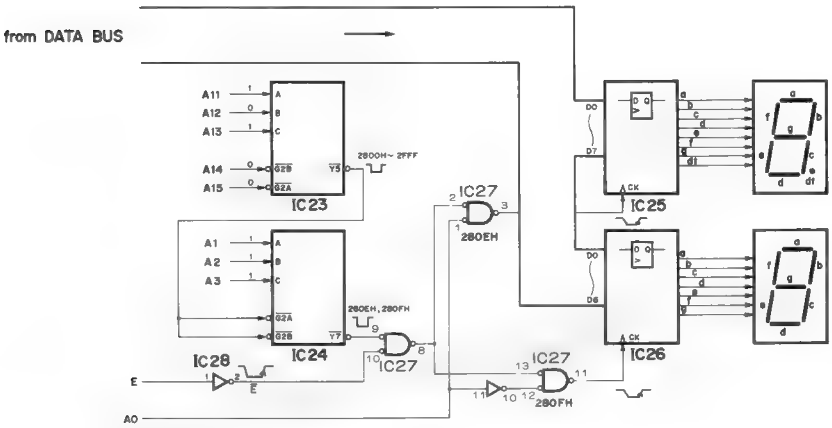
The memory protect switch is connected to the $\overline{\text{WE}}$ terminal of the EEPROM, and turning the switch to the "ON" position will connect the $\overline{\text{WE}}$ terminal to the terminal PC6 of the PPI8225. Since the $\overline{\text{WE}}$ terminal is pulled up to +5V by a resistor R2, the logic level at the terminal PC6 is set to be H, determined by a ratio of R1/(R1 + R2) (normally, the logic level is maintained at L since terminal PC6 is connected to the ground through the resistor R1). The main CPU detects that the protect switch is ON, and thus the EEPROM is protected with software. Also, the $\overline{\text{WE}}$ terminal is mechanically isolated from the R/$\overline{\text{W}}$ terminal and is pulled up to the +5V through the resistor R2 thus prohibiting WRITE operations. READ operations are possible since the $\overline{\text{OE}}$ of the IC2 is selected for reading data from the EEPROM. When the protect switch is switched to "OFF", the READ/WRITE operations follow the movement of the R/W terminal, i. e. logic level H corresponds to READ operations and L to WRITE operations. The main CPU detects that the protect switch is "OFF", since terminal PC6 of the PPI8255 is set to logic level L through the resistor R1 connected to the ground.

REMARKS

4) 7 Segment LED Driving Circuit

The LED (Light Emitting Diode) driving circuits (IC25, 26) latch the data for display on the data bus using the trailing edge of the E; Enable signal (or the leading edge of the E signal) D flip-flops (IC25, 26) are used for latching data and are triggered by the leading edge.

LED DRIVE CIRCUIT



See figure V–8 (Page 80)

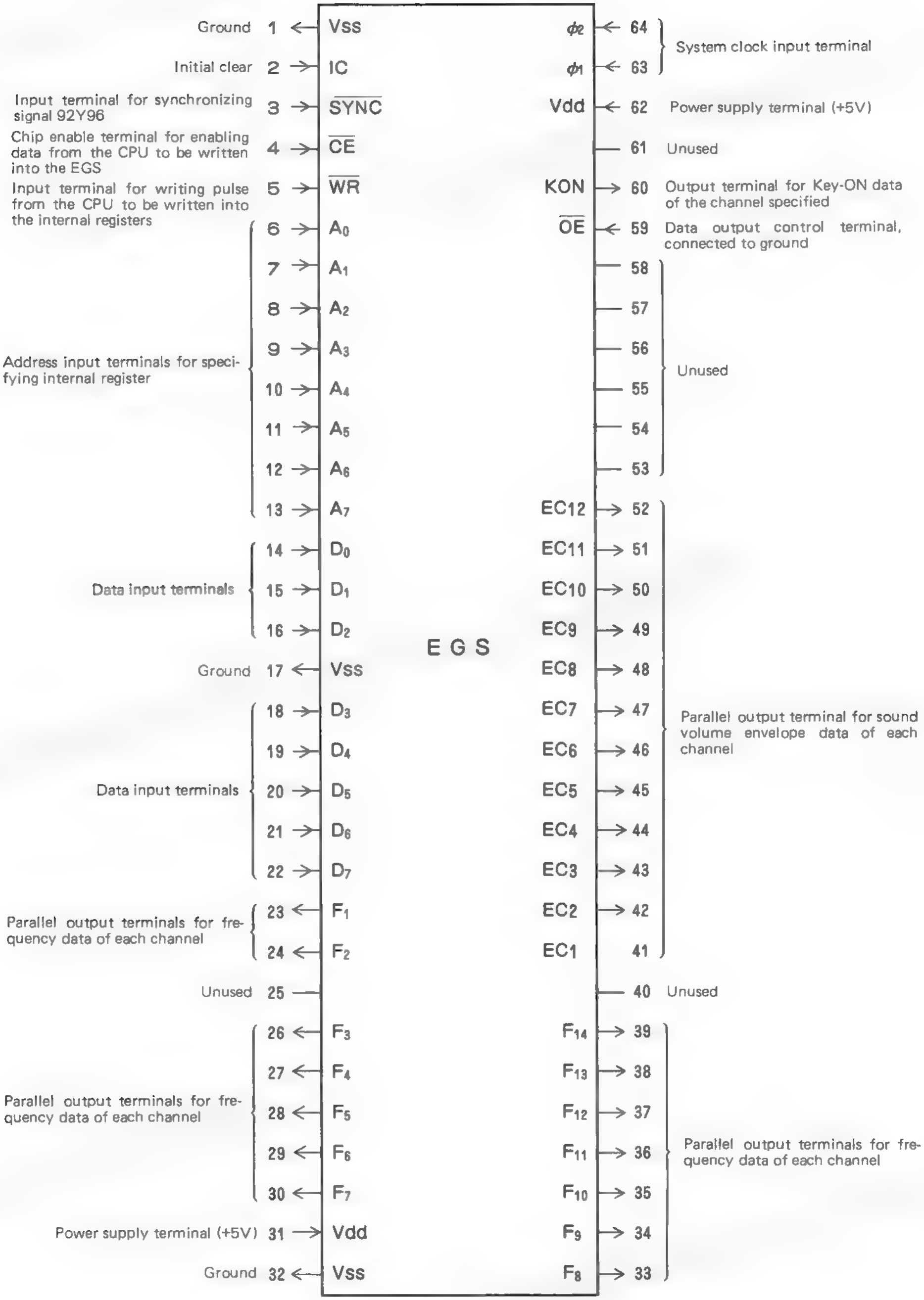
The $\overline{Y_5}$ of the IC23 becomes low when memory locations 2800H ~ 2FFFH are selected and hence IC24 also becomes enabled.

The LEDs accept display data only when $\overline{Y_7}$ of IC24 becomes low. Since the $\overline{Y_7}$ becomes low when the A, B, and C terminals of the IC24 simultaneously become high, i. e. $\overline{Y_7}$ becomes low only when 280EH and 280FH are addressed. Due to the result of the AND operation between the $\overline{Y_7}$ and inverted E signal, the output voltage at the pin number 8 of the IC27 becomes low when signal E is maintained high (or E is low). The result of the AND operation between pin number 8 of the IC27 and the address line A_0 (lowest bit) is then used for selecting either the upper digit LED(280EH) when A_0 is low or the lower digit LED(280FH) when A_0 is high.

The display on the LEDs immediately after the power is turned on has nothing to do with the operation of the CPU. It shows unstable data before the circuit becomes stable. However, units produced after serial #21836 have been made to display "88"

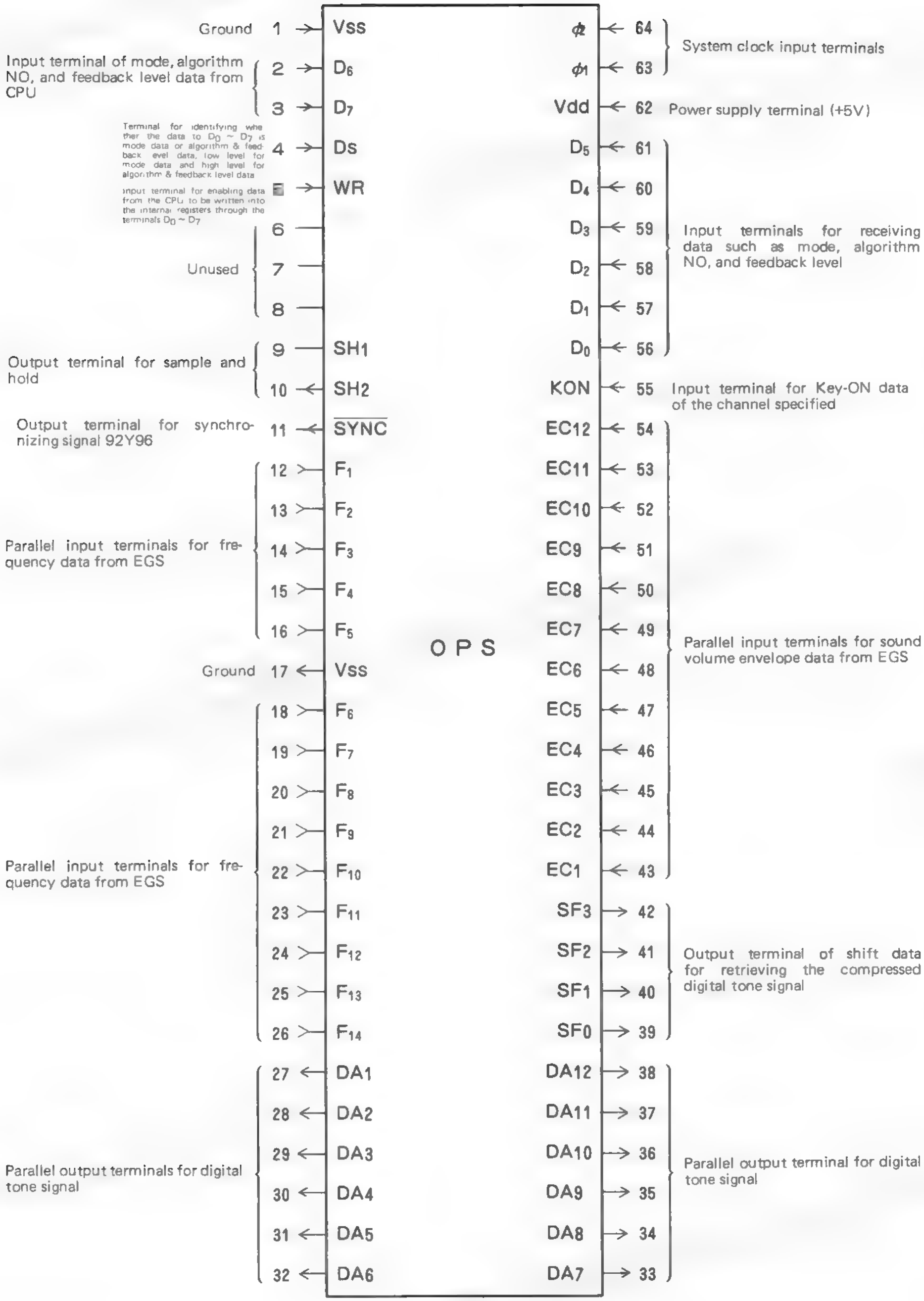
5) FM Tone Generator

The FM Tone Generator consists of the EGS (the Envelope Generator of the Synthesizer) and OPS (the Operator of the Synthesizer).
The EGS creates, based on the data transmitted from the main CPU and in synchronization with the synchronizing signal 92Y96, the sound volume envelope data and frequency data of each channel for a total of 96 channels and outputs them along with the key on(KON) data which assigned to each channel number.

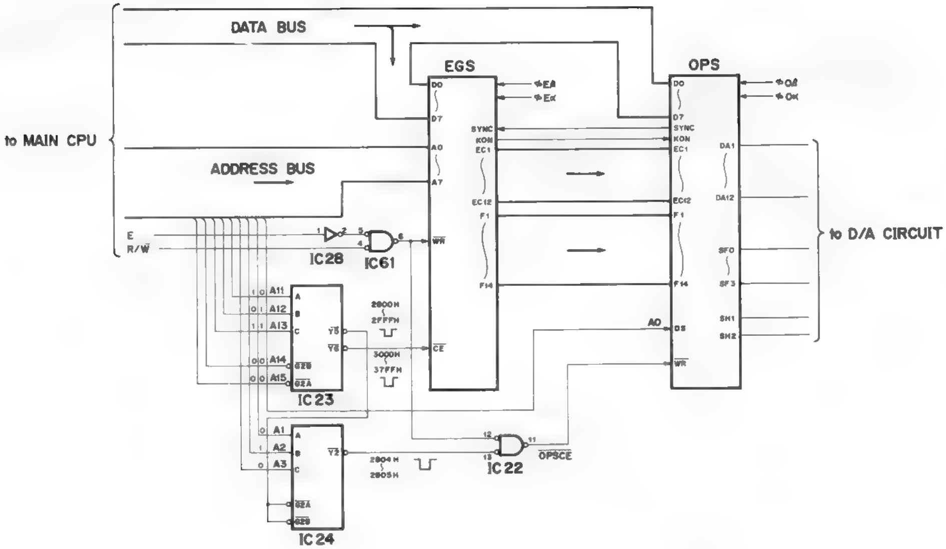

E G S
Ground 1 Vss
Initial clear 2 IC
Input terminal for synchronizing signal 92Y96 3 SYNC
Chip enable terminal for enabling data from the CPU to be written into the EGS 4 CE
Input terminal for writing pulse from the CPU to be written into the internal registers 5 WR
Address input terminals for specifying internal register
6 A0
7 A1
8 A2
9 A3
10 A4
11 A5
12 A6
13 A7
Data input terminals
14 D0
15 D1
16 D2
Ground 17 Vss
Data input terminals
18 D3
19 D4
20 D5
21 D6
22 D7
Parallel output terminals for frequency data of each channel
23 F1
24 F2
Unused 25
Parallel output terminals for frequency data of each channel
26 F3
27 F4
28 F5
29 F6
30 F7
Power supply terminal (+5V) 31 Vdd
Ground 32 Vss
φ2 64
φ1 63
System clock input terminal
Vdd 62 Power supply terminal (+5V)
61 Unused
KON 60 Output terminal for Key-ON data of the channel specified
OE 59 Data output control terminal, connected to ground
58
57
56
55
54
53
Unused
EC12 52
EC11 51
EC10 50
EC9 49
EC8 48
EC7 47
EC6 46
EC5 45
EC4 44
EC3 43
EC2 42
EC1 41
Parallel output terminal for sound volume envelope data of each channel
40 Unused
F14 39
F13 38
F12 37
F11 36
F10 35
F9 34
F8 33
Parallel output terminals for frequency data of each channel

The OPS performs FM (Frequency Modulation) arithmetic operations based on the sound volume envelope data and frequency data of each channel, and the KON data and according to the algorithm NO currently stored within the OPS and feed back level. And then it will generate tone signal made up of 12bits of digital data. The output data from the OPS is 12bits but is equivalent to 14 to 16bits by compressing data at the low level by 2, 4 of 8. The compressed data can be retrieved by using shift data (SF0 ~ SF3).

SF0. . . . . . . Expansion OFF
SF1. . . . . . . Expansion ON (*Double)
SF2. . . . . . . Expansion ON (Quadruple)
SF3. . . . . . . Expansion ON (Eight times)


Ground
1 Vss
φ2 64
φ1 63
System clock input terminals
Input terminal of mode, algorithm NO, and feedback level data from CPU
2 D6
3 D7
Vdd 62
Power supply terminal (+5V)
Terminal for identifying whether the data to D0 ~ D7 is mode data or algorithm & feedback level data, low level for mode data and high level for algorithm & feedback level data
4 Ds
D5 61
Input terminal for enabling data from the CPU to be written into the internal registers through the terminals D0 ~ D7
5 WR
D4 60
Unused
6
7
8
D3 59
D2 58
D1 57
D0 56
Input terminals for receiving data such as mode, algorithm NO, and feedback level
Output terminal for sample and hold
9 SH1
10 SH2
KON 55
Input terminal for Key-ON data of the channel specified
Output terminal for synchronizing signal 92Y96
11 SYNC
EC12 54
Parallel input terminals for frequency data from EGS
12 F1
13 F2
14 F3
15 F4
16 F5
EC11 53
EC10 52
EC9 51
EC8 50
EC7 49
OPS
Ground
17 Vss
EC6 48
Parallel input terminals for sound volume envelope data from EGS
Parallel input terminals for frequency data from EGS
18 F6
19 F7
20 F8
21 F9
22 F10
23 F11
24 F12
25 F13
26 F14
EC5 47
EC4 46
EC3 45
EC2 44
EC1 43
SF3 42
SF2 41
SF1 40
SF0 39
Output terminal of shift data for retrieving the compressed digital tone signal
Parallel output terminals for digital tone signal
27 DA1
28 DA2
29 DA3
30 DA4
31 DA5
32 DA6
DA12 38
DA11 37
DA10 36
DA9 35
DA8 34
DA7 33
Parallel output terminal for digital tone signal

1) EGS Operation



REMARKS

See figures VI–1 ~ VI–5 (Pages 81 ~ 83)

It is called "Image" that is the same memories are specified by specifying different addresses as shown in the example at right.

Since the EGS is enabled by the $\overline{Y_6}$ of the decoder IC23, the addresses 3000H ~ 37FFH (2Kbytes) are assigned to the EGS. It seems that resistors of 2Kbytes are built into the EGS. However, actual address lines are 8, A_0 ~ A_7 and the amount of addresses to be accessed by using 8 address lines is only 256bytes as shown in the following expression.

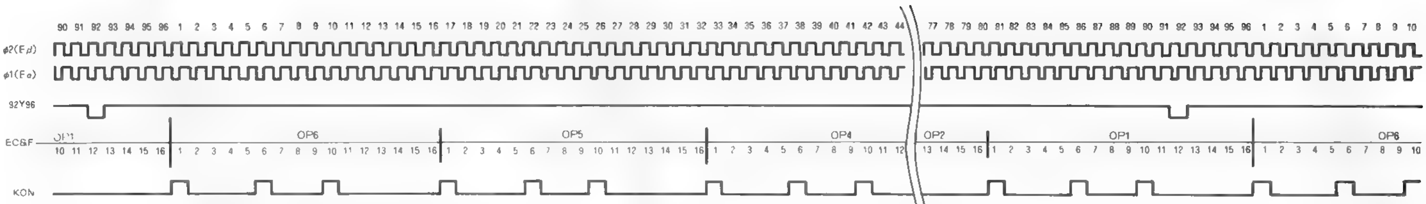
2^8 = 256bytes

Thus, the following 8 sets of 256bytes will correspond to the same register with 256 bytes within the EGS.

3000H ~ 30FFH 256bytes
3100H ~ 31FFH 256bytes
3200H ~ 32FFH 256bytes
3300H ~ 33FFH 256bytes
3400H ~ 34FFH 256bytes
3500H ~ 35FFH 256bytes
3600H ~ 36FFH 256bytes
3700H ~ 37FFH 256bytes

The result of the AND operation between the inverted E signal and the R/$\overline{W}$ signal is fed to the $\overline{WR}$ terminal of the EGS to allow data to be written into the internal registers. The data is written into the registers by the leading edge of the E signal.

EGS OUTPUT TIMING CHART



The data of each channel is output in synchronization with the leading edge of the signal 0 2. This timing chart shows that the channels 1, 6, and 10 are generating sounds as well as the next 92Y96 cycle. The EC(12bits) and F(14bits) are output simultaneously for each channel.

The following shows the output timing chart of the EGS.

REMARKS

See figures VI–1 to VI–5 (Pages 81, 82) and VI–15 to VI–18 (Page 86)

2) OPS Operation

The OPS receives the following 2bytes of data directly from the CPU.

- **MODE (operation mode of OPS). 1byte**
- **ALGORITHM NO (upper 5bits)**
- **FEEDBACK LEVEL (lower 3bits). 1byte**

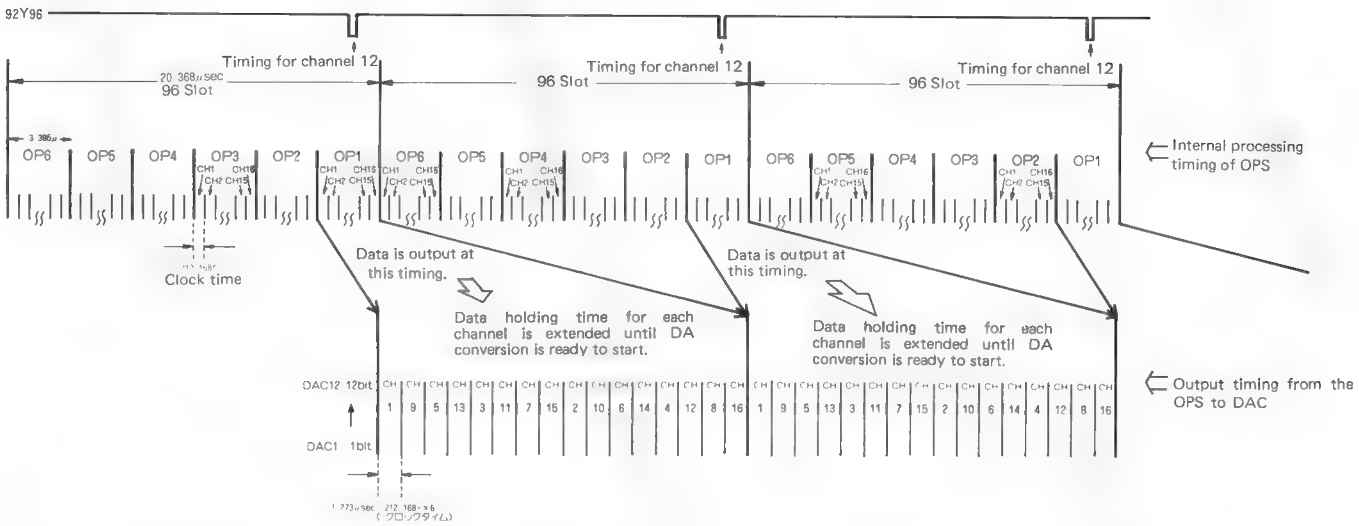
The result of the AND operation between the $\overline{Y2}$ terminal of the address decoder IC24 and the result of the AND operation between the $\overline{E}$ signal and the $R/\overline{W}$ signal is fed to the $\overline{WR}$ terminal of the OPS to allow the above two bytes of data to be written into the OPS. The addresses assigned are 2804H and 2805H.

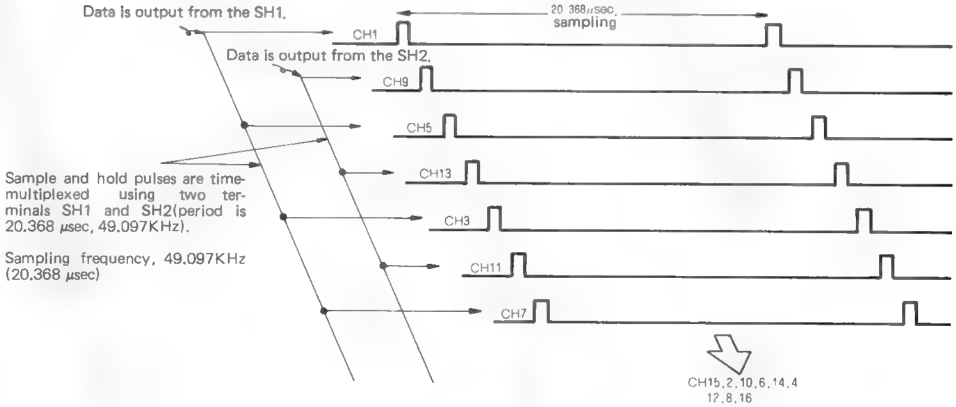
Since the address line A_0 is connected to data set terminal DS of the OPS, the addresses 2804H and 2805H are assigned in the following way.

A_0 low . . . . . the address 2804H is used for specifying the mode register of the OPS

A_0 high . . . . the address 2805H is used for specifying the data registers (ALGORITHM NO, FEEDBACK LEVEL) of the OPS

The following shows the output timing chart of the OPS.

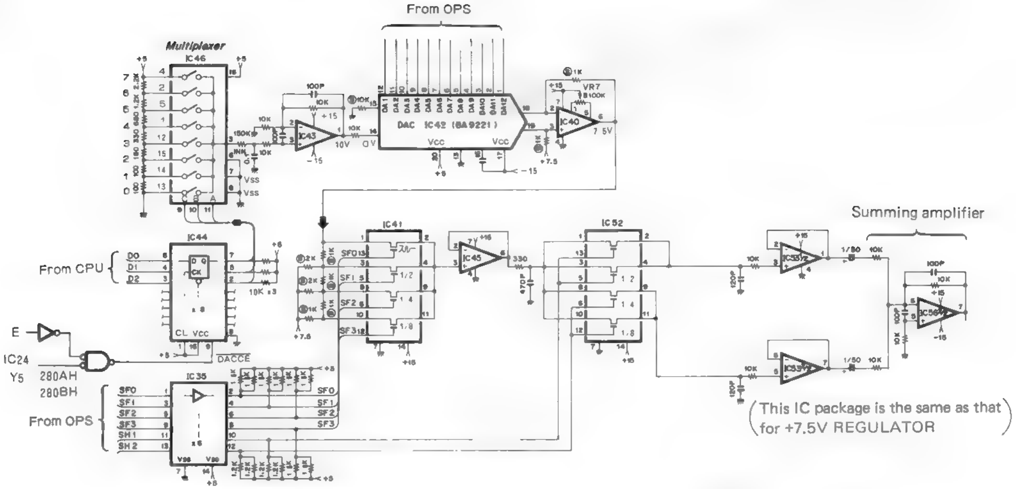




Sample and hold pulses are time-multiplexed using two terminals SH1 and SH2(period is 20.368 μsec, 49.097KHz).

Sampling frequency, 49.097KHz (20.368 μsec)

6) D/A Converter



(This IC package is the same as that for +7.5V REGULATOR)

REMARKS

See figures VII–1 ~ VII–4 (Page 87)

The 12 bits of digital data fed from the OPS is input to the DAC IC42 and the maximum output current (the maximum output voltage for the output terminal of the inverting amplifier IC40) is determined by the current fed to the pin14 of the DAC IC42 through the resistor (10K ohms).

The current fed to this terminal is selected by the multiplexer: IC46.

Normally pin4 of the multiplexer is selected and the voltage observed at the pin3 (output terminal of the multiplexer is DC +5V. Also, the voltage at output pin1 of the operational amplifier IC43 is 10V.

This multiplexer plays the role of electronic level controller when used with the KX1. Sound level can be controlled by the lowermost 3bits of the data at location 280AH or 280BH.

Normally, inputs D0, D1, and D2 to the IC44 are selected to be 1(D0), 1(D1), and 1(D2) to allow a voltage of 5V to be observed at the multiplexer output. When KX1 is connected to DX7, setting D0, D1, and D2 to 0, 0, and 0 respectively will result in a multiplexer output voltage of 0. Thus no sound is heard from the speaker under this situation.

The shift data SF0 to SF3 are determined by the level of the digital data fed to the DAC and the output timing of the shift data is the same as that of the digital data.

Concretely speaking. The shift data SF0 to SF3 are determined as follows.

SF0 (logic level 0)..... no shifting was carried out within the OPS regarding the data input to the DAC.

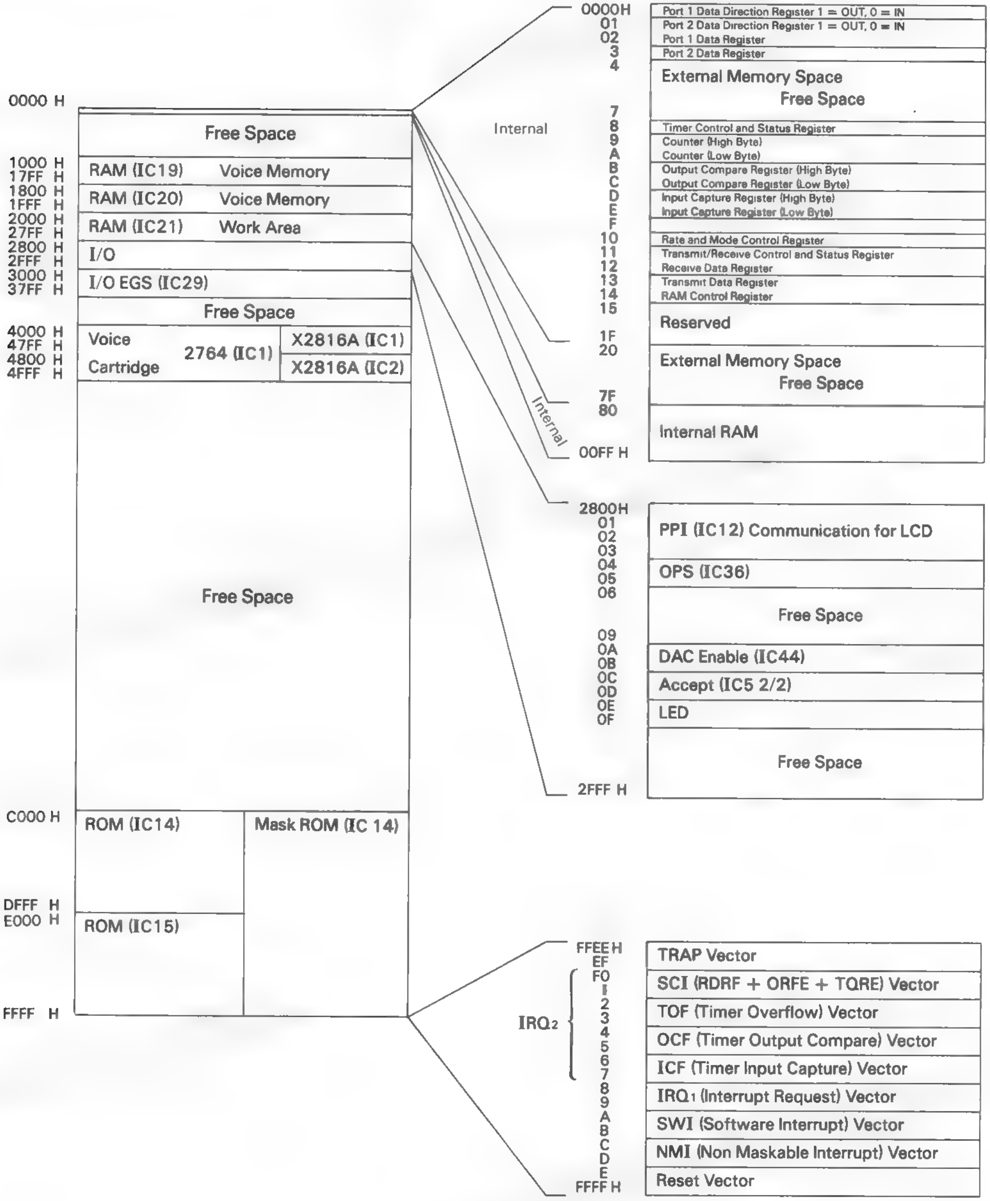
SF1 (logic level 1)..... one bit shift left was carried out within the OPS regarding the data input to the DAC (or the data was multiplied by 2).

SF2 (logic level 2)..... 2bits shift left was carried out within the OPS regarding the data input to the DAC (or the data was multiplied by 4).

SF3 (logic level 3)..... 3bits shift left was carried out within the OPS regarding the data input to the DAC (or the data was multipled by 8).

7) Memory Map

The following shows the memory map of the entire DX7 system.



0000 H	Free Space	
1000 H 17FF H	RAM (IC19)	Voice Memory
1800 H 1FFF H	RAM (IC20)	Voice Memory
2000 H 27FF H	RAM (IC21)	Work Area
2800 H 2FFF H	I/O	
3000 H 37FF H	I/O EGS (IC29)	
	Free Space	
4000 H 47FF H	Voice Cartridge 2764 (IC1)	X2816A (IC1)
4800 H 4FFF H		X2816A (IC2)
	Free Space	
C000 H	ROM (IC14)	Mask ROM (IC 14)
DFFF H E000 H	ROM (IC15)	
FFFF H		

Internal

0000H	Port 1 Data Direction Register 1 = OUT, 0 = IN
01	Port 2 Data Direction Register 1 = OUT, 0 = IN
02	Port 1 Data Register
3	Port 2 Data Register
4 7	External Memory Space Free Space
8	Timer Control and Status Register
9	Counter (High Byte)
A	Counter (Low Byte)
B	Output Compare Register (High Byte)
C	Output Compare Register (Low Byte)
D	Input Capture Register (High Byte)
E	Input Capture Register (Low Byte)
F	
10	Rate and Mode Control Register
11	Transmit/Receive Control and Status Register
12	Receive Data Register
13	Transmit Data Register
14	RAM Control Register
15 1F	Reserved
20 7F	External Memory Space Free Space
80 00FF H	Internal RAM

Internal

2800H 01	PPI (IC12) Communication for LCD
02 03	OPS (IC36)
04 05 06 09	Free Space
0A 0B	DAC Enable (IC44)
0C 0D	Accept (IC5 2/2)
0E 0F	LED
2FFF H	Free Space

FFEE H EF	TRAP Vector
F0 1	SCI (RDRF + ORFE + TDRE) Vector
2 3	TOF (Timer Overflow) Vector
4 5	OCF (Timer Output Compare) Vector
6 7	ICF (Timer Input Capture) Vector
8 9	IRQ1 (Interrupt Request) Vector
A B	SWI (Software Interrupt) Vector
C D	NMI (Non Maskable Interrupt) Vector
E FFFF H	Reset Vector

IRQ2

Reference EGS Address Map

	bit /1 DATA	Number of bytes	
LF	14	16CH x 2	Key Code log F
RF	14	6 op. x 2	Ratio of Frequency
DT	4	6 op.	Detune
R	6	6 op. x 4 seg	Rate
L	6	6 op. x 4 seg	Level
TL	8	6 op. x 16CH	Total Level
RS, MS	3 + 2	6 op.	Rate Scaling Mod. Sensitivity
AMOD	8	1	Amplitude Mod.
KOF	1	1	Key On, Off
PMOD	12	1	Pitch Mod.

Total 203 byte

Address		
$00	LOG F	CH1
01		
02		CH2
03		
04		CH3
05		
06		CH4
07		
08		CH5
09		
0A		CH6
0B		
0C		CH7
0D		
0E		CH8
0F		
10		CH9
11		
12		CH10
13		
14		CH11
15		
16		CH12
17		
18		CH13
19		
1A		CH14
1B		
1C		CH15
1D		
1E		CH16
1F		

LF14	LF13	LF12	LF11	LF10	LF9	LF8	LF7
LF6	LF5	LF4	LF3	LF2	LF1	X	X

LF1 ~ 14	The value which each bit has (cent)
14	9600
13	4800
12	2400
11	1200
10	600
9	300
8	150
7	75
6	37.5
5	18.8
4	9.4
3	4.7
2	2.3
1	1.2

LF { Key code
PEG RAM
Transpose RAM

EVENT

NOTE: EVENT when Key event occurred
DATA when Panel DATA was changed
TIME At every certain time interval

20	LOG RF	OP6
21		
22		OP5
23		
24		OP4
25		
26		OP3
27		
28		OP2
29		
2A		OP1
2B		

RF14	RF13	RF12	RF11	RF10	RF9	RF8	RF7
RF6	RF5	RF4	RF3	RF2	RF1	X	0

FIXED = 0

RF14	RF13	RF12	RF11	RF10	RF9	RF8	RF7
RF6	RF5	RF4	RF3	RF2	RF1	X	1

FIXED = 1

The value that each bit has is the same as that of LF.

DATA

RF = Ratio of Frequency

30	DETUNE	OP6
31		OP5
32		OP4
33		OP3
34		OP2
35		OP1

X	X	X	X	FD4	FD3	FD2	FD1

(SIGN)
0 +
1 −

DATA

40	R	OP6	R1
41			R2
42			R3
43			R4
44		OP5	R1
45			R2
46			R3
47			R4
48		OP4	R1
49			R2
4A			R3
4B			R4
4C		OP3	R1
4D			R2
4E			R3
4F			R4
50		OP2	R1
51			R2
52			R3
53			R4
54		OP1	R1
55			R2
56			R3
57			R4

X	X	R6	R5	R4	R3	R2	R1

Each rate is expressed with 6bits.

DATA

60	L	OP6	L1
61			L2
62			L3
63			L4
64		OP5	L1
65			L2
66			L3
67			L4
68		OP4	L1
69			L2
6A			L3
6B			L4
6C		OP3	L1
6D			L2
6E			L3
6F			L4
70		OP2	L1
71			L2
72			L3
73			L4
74		OP1	L1
75			L2
76			L3
77			L4

X	X	L6	L5	L4	L3	L2	L1

Each level is expressed with 6bits.

DATA

80	TL	OP6	CH1
81			CH2
82			CH3
83			CH4
84			CH5
85			CH6
86			CH7
87			CH8
88			CH9
89			CH10
8A			CH11
8B			CH12
8C			CH13
8D			CH14
8E			CH15
8F			CH16
90		OP5	CH1
91			CH2
92			CH3
93			CH4
94			CH5
95			CH6
96			CH7
97			CH8
98			CH9
99			CH10
9A			CH11
9B			CH12
9C			CH13
9D			CH14
9E			CH15
9F			CH16
A0		OP4	CH1
A1			CH2
A2			CH3
A3			CH4
A4			CH5
A5			CH6
A6			CH7
A7			CH8
A8			CH9
A9			CH10
AA			CH11
AB			CH12
AC			CH13
AD			CH14
AE			CH15
AF			CH16

TL8	TL7	TL6	TL5	TL4	TL3	TL2	TL1

TL1 ~ 8	The value that each bit has.
TL8	48
7	24
6	12
5	6
4	3
3	1.5
2	0.75
1	0.375

EVENT

B0	TL	OP3	CH1
B1			CH2
B2			CH3
B3			CH4
B4			CH5
B5			CH6
B6			CH7
B7			CH8
B8			CH9
B9			CH10
BA			CH11
BB			CH12
BC			CH13
BD			CH14
BE			CH15
BF			CH16
C0		OP2	CH1
C1			CH2
C2			CH3
C3			CH4
C4			CH5
C5			CH6
C6			CH7
C7			CH8
C8			CH9
C9			CH10
CA			CH11
CB			CH12
CC			CH13
CD			CH14
CE			CH15
CF			CH16
D0		OP1	CH1
D1			CH2
D2			CH3
D3			CH4
D4			CH5
D5			CH6
D6			CH7
D7			CH8
D8			CH9
D9			CH10
DA			CH11
DB			CH12
DC			CH13
DD			CH14
DE			CH15
DF			CH16

TL8	TL7	TL6	TL5	TL4	TL3	TL2	TL1

E0	S	OP6
E1		OP5
E2		OP4
E3		OP3
E4		OP2
E5		OP1

X	X	X	MS2	MS1	RS3	RS2	RS1

DATA

F0	AMOD	Common to all OPs.

MS = 3 (max.)

dB 24	12	6	3	1.5	0.75	0.375	0.188
EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1

After Touch
BC, LFO

TIME

F1	KOF	

X	X	CH4	CH3	CH2	CH1	1	0	KEY OFF
X	X	CH4	CH3	CH2	CH1	0	1	KEY ON

CH1 ~ CH16 are expressed with these 4bits. EVEN

F2	PMOD	Common to all OPs.
F3		

Cent Sign	1200	600	300	150	75	37.5	18.8
FM12	FM11	FM10	FM9	FM8	FM7	FM6	FM5
FM4	FM3	FM2	FM1	X	X	X	X
9.4	4.1	2.3	1.2				

Pitch Bend
LFO
(= 2 oct.)

TIME

FM12 ~ FM1 2's compl.

I -1 Sub CPU

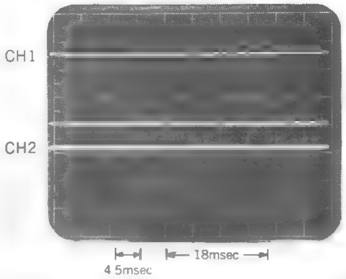
Check Point

CH1 Inverted signal of B7 (19pin)

A/D (6pin) START (2pin) ALE

CH2 B5 17pin

IC3 (19pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -2 Sub CPU

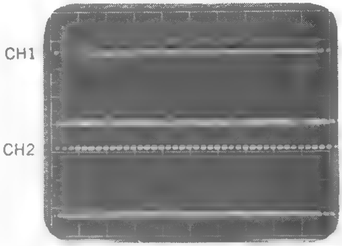
Check Point

CH1 Inverted signal of B7 (19pin)

A/D (6pin) START (22pin) ALE

CH2 B4 16pin

IC4 (19pin)



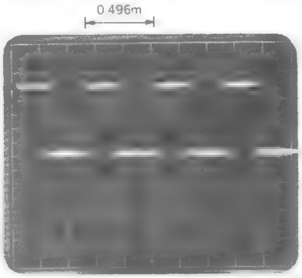
CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -3 Sub CPU

Check Point

The signal on B4 (16pin) was expanded on time axis.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.2mS TIME DIV	
AC·DC	AC·DC

Probe 10 : 1

I -4 Sub CPU

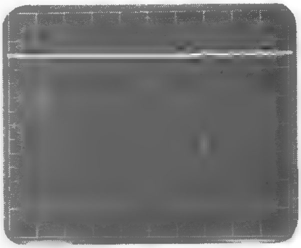
Check Point

Inverted signal of C0 (8pin)

Conditions

Storage oscilloscope was used.

The pulse width is 0.1



CH.1	CH.2
VOLT DIV	VOLT DIV
10mS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

I -5 Sub CPU

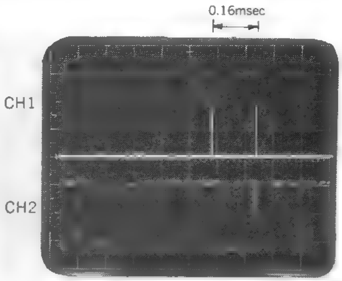
Check Point

CH1	C1 (9pin)
CH2	C2 BUSY

Conditions

Storage oscilloscope was used.

One key was kept depressed.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.1mS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

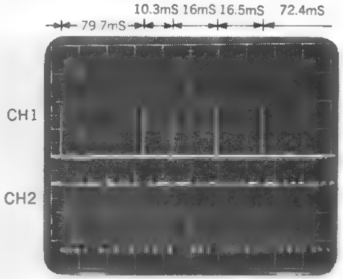
I -6 Sub CPU

Check Point

In the above picture (I–5 Sub CPU), the four keys were kept depressed.

Conditions

Storage oscilloscope was used.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
10mS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

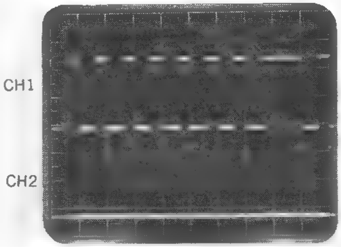
I -7 A/D

Check Point

IC11 A/D converter

CH1 EOC (7pin)

CH2 OE (9pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

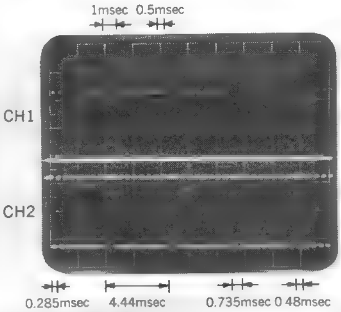
I -8 A/D

Check Point

IC12 A/D converter

CH1 START (6pin)

CH2 EOC (7pin)



CH.1	CH.2
0.2V VOLT DIV	2V VOLT DIV
2mS TIME DIV	
AC·DC	AC·DC

CHOP mode

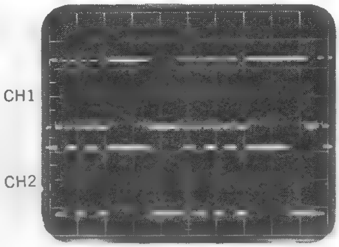
Probe 10 : 1

I -9 SWITCH SCAN

Check Point

IC1

CH1	A (1pin)
CH2	B (2pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.1mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

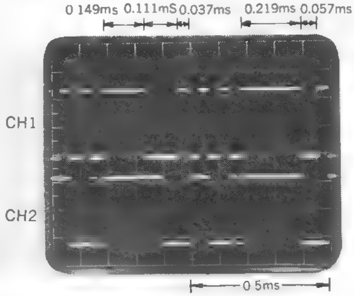
Probe 10 : 1

I -10 SWITCH SCAN

Check Point

IC1

CH1	B (2pin)
CH2	C (3pin)



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.1mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -11 SWITCH SCAN

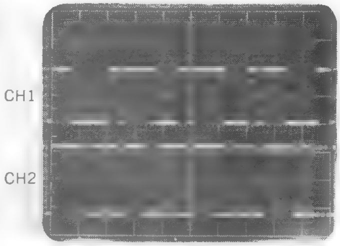
Check Point

CH1 IC1 $\overline{G2A}$/$\overline{G2B}$ (4,5pin) or

IC2 G1 (6pin)

CH2 IC1 A (1pin)

HOLD OFF function used



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

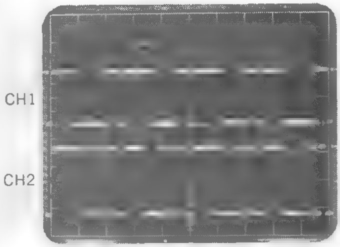
I -12 SWITCH SCAN

Check Point

CH1 IC1 $\overline{G2A}$/$\overline{G2B}$ (4,5pin)

IC2 G1 (6pin)

CH2 IC1 B (2pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

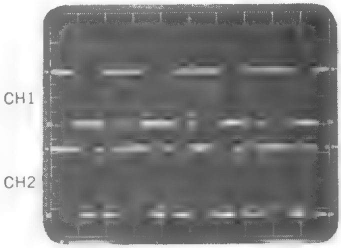
I -13 SWITCH SCAN

Check Point

CH1 IC1 $\overline{G2A}$/$\overline{G2B}$ (4,5pin)

IC2 G1 (6pin)

CH2 IC1 C (3pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

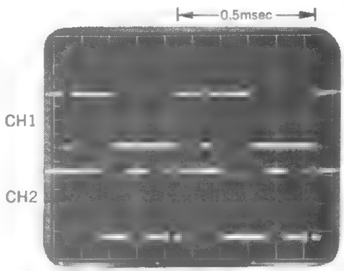
I -14 SWITCH SCAN

Check Point

CH1 IC1 $\overline{G2A}$/$\overline{G2B}$ (4,5pin)

IC2 G1 (6pin)

CH2 IC1 C (3pin)



Upper picture expanded

CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

All these pictures show the output signal from the SUB CPU.

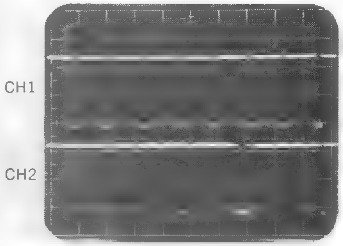
I -15 SWITCH SCAN

Check Point

IC1

CH1 $\mathrm{C^{\#}\overline{Y0}}$ (15pin)

CH2 $\mathrm{D\overline{Y1}}$ (14pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

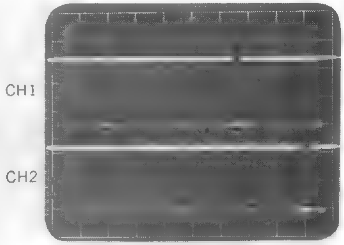
I -16 SWITCH SCAN

Check Point

IC1

CH1 $\mathrm{D\overline{Y1}}$ (14pin)

CH2 $\mathrm{D^{\#}\overline{Y2}}$ (13pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

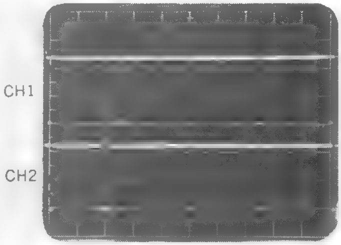
I -17 SWITCH SCAN

Check Point

IC1

CH1 $\mathrm{D^{\#}\overline{Y2}}$ (13pin)

CH2 $\mathrm{E\overline{Y3}}$ (12pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

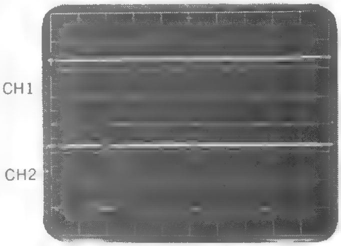
I -18 SWITCH SCAN

Check Point

IC1

CH1 $\mathrm{E\overline{Y3}}$ (12pin)

CH2 $\mathrm{E\overline{Y4}}$ (11pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DiV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

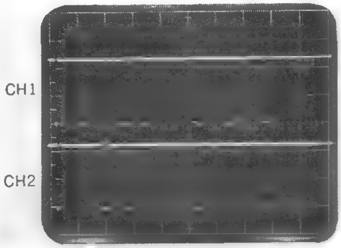
I -19 SWITCH SCAN

Check Point

IC1

CH1 E$\overline{Y4}$ (11pin)

CH2 F#$\overline{Y5}$ (10pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

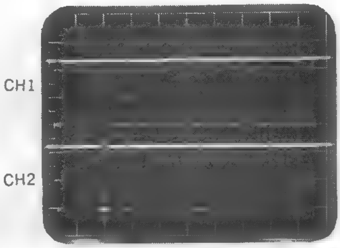
I -20 SWITCH SCAN

Check Point

IC1

CH1 F#$\overline{Y5}$ (10pin)

CH2 G$\overline{Y6}$ (9pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

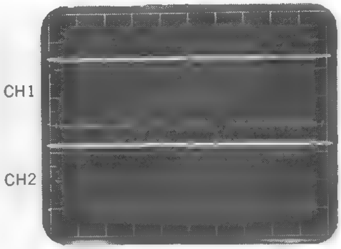
I -21 SWITCH SCAN

Check Point

IC1

CH1 G$\overline{Y6}$ (9pin)

CH2 G#$\overline{Y7}$ (7pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

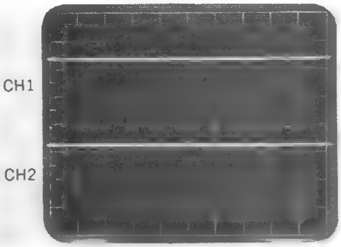
I -22 SWITCH SCAN

Check Point

IC2

CH1 $A\overline{Y0}$ (pin15)

CH2 $A^{\#}\overline{Y1}$ (pin14)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -23 SWITCH SCAN

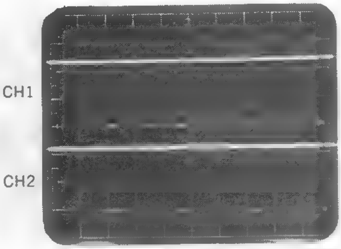
Check Point

IC2

CH1 $A^{\#}\overline{Y1}$ (pin14)

CH2 $B\overline{Y2}$ (pin13)

$C\overline{Y3}$; $CL\overline{Y4}$ — same pattern



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

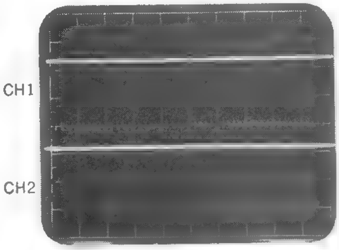
I -24 SWITCH SCAN

Check Point

IC2

CH1 $CL\overline{Y4}$ (pin11)

CH2 $SA\overline{Y5}$ (pin10)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

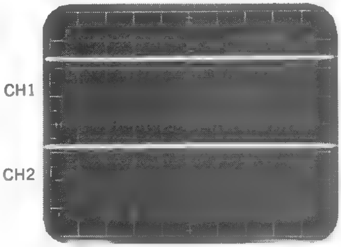
I -25 SWITCH SCAN

Check Point

IC2

CH1 $SA\overline{Y5}$ (pin10)

CH2 $SB\overline{Y6}$ (pin7)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

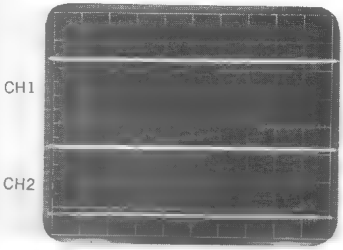
Probe 10 : 1

I -26 SWITCH SCAN

Check Point

IC2

CH1	SB$\overline{\text{Y6}}$ (9pin)
CH2	SC$\overline{\text{Y7}}$ (7pin)



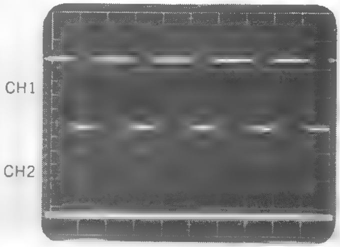
CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -27 SWITCH SCAN

Check Point

The signal $\overline{\text{Y7}}$ in the above picture (I–26 SWITCH SCAN) was expanded on the time axis.



The pulse width is 418μsec.

CH.1	CH.2
0.2V /DIV	0.2V /DIV
0.2mS /DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -28 SWITCH SCAN

Check Point

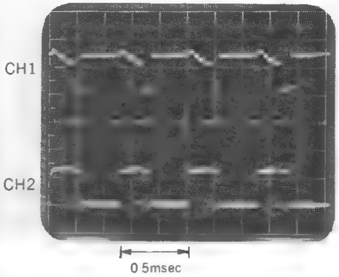
IC3/IC4

CH1 output signals on 3, 5, 7, 9, 12 pins

CH2 output signal on 14pin

Conditions

SWITCH scan output where no keys are being depressed.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

I -29 SWITCH SCAN

Check Point

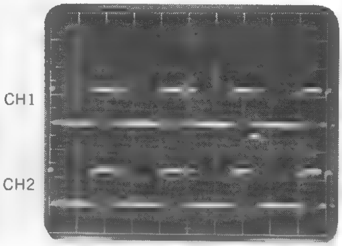
IC3/IC4

CH1 output signal on 16pin

CH2 output signal on 18pin

Condition

SWITCH scan output where no keys are being depressed.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.2mS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

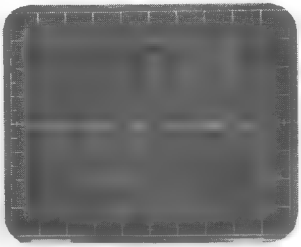
II-1 CLOCK

Check point

IC51 (2pin)

(Inverter output)

CLOCK 37.706MHz



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.02μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

II-2 CLOCK

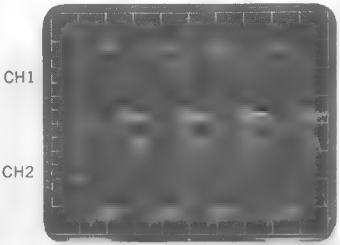
Check point

CH1 φEα, 0α

IC58,59 (2pin)

CH2 φEβ, 0β

IC58,59 (3pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.1μS /DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

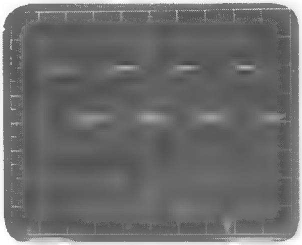
Distorted signal waveform is observed here because of improper grounding at the time the picture was taken.

II-3 CLOCK

Check Point

IC48 (5pin)

φM 4.71325MHz



CH.1	CH 2
0.2V /DIV	/DIV
0.1μS /DIV	
AC [✓ DC]	AC·DC

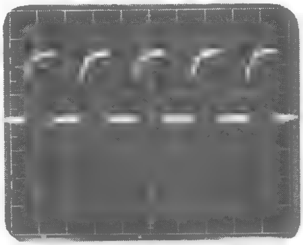
Probe 10 : 1

II-4 CLOCK

Check Point

IC5 (9pin)

Timer CLOCK 250KHz



CH.1	CH.2
0.2V /DIV	/DIV
2μS /DIV	
AC [✓ DC]	AC·DC

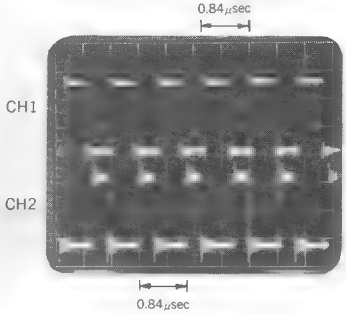
Probe 10 : 1

III-1 MAIN CPU

Check Point

CH1 E (40pin)

CH2 ALE (39pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

III-2 MAIN CPU

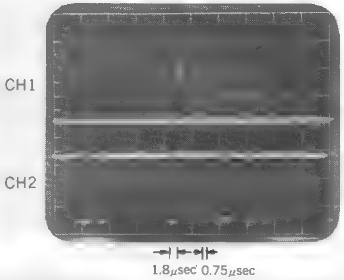
Check Point

CH1 ALE (39pin)

CH2 R/$\overline{\text{W}}$ (38pin)

Conditions

Storage oscilloscope was used.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

III-3 MAIN CPU

Check Point

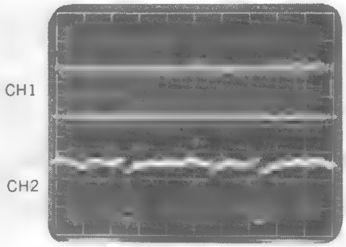
IC61

CH1 $\overline{\text{E}}$ (5pin)

CH2 $\overline{\text{WR}}$ (6pin)

Conditions

Storage oscilloscope was used.



CH.1	CH.2
0.2V VOLT DIV	0 2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

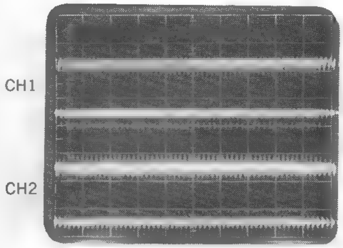
III-4 MAIN CPU

Check Point

IC61

CH1 $\overline{E}$ (5pin)

CH2 $\overline{RD}$ (8pin)



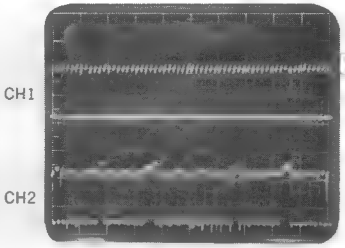
CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

III-5 MAIN CPU

Check Point

Storage oscilloscope was used instead of normal oscilloscope in the above picture (III–4 MAIN CPU).



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

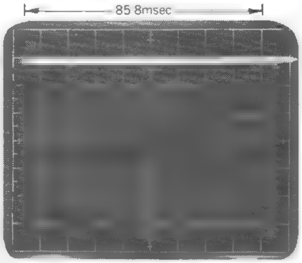
Probe 10 : 1

III-6 MAIN CPU

Check Point

P24 (12pin)

High logic level on P23 (11pin)

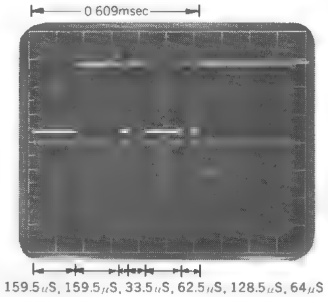


CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10mS TIME DIV	
AC [✓ DC]	AC·DC

III-7 MAIN CPU

Check Point

The above picture (III–6 MAIN CPU) was expanded on the time axis.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.1mS TIME/DIV	
AC [✓ DC]	AC·DC

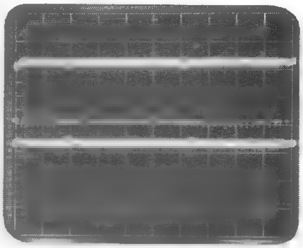
Probe 10 : 1

IV-1 MEMORY

Check Point

CH1 IC22 (8pin)
RAM IC21 $\overline{CE1}$

CH2 IC23 (10pin)
Enable signal of IC24
(Address decoder enable of OUTPORT)



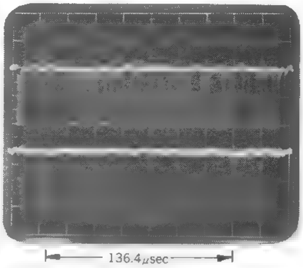
CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
20μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

IV-2 MEMORY

Check Point

Storage oscilloscope was used.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
20μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

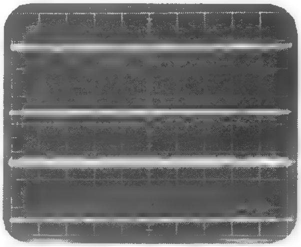
Probe 10 : 1

IV-3 MEMORY

Check Point

CH1 ROM IC14 $\overline{CE1}$

CH2 ROM IC15 $\overline{CE1}$



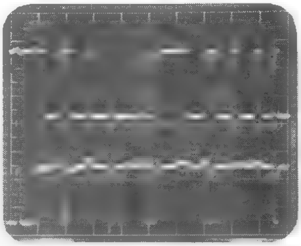
CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT DIV
2mS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

IV-4 MEMORY

Check Point

Storage oscilloscope was used.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

IV-5 MEMORY

Check Point

$\overline{\text{CH1}}$ RAM IC19

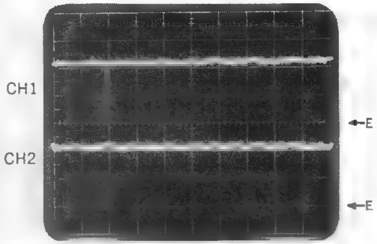
where switches of CE1 1 ~16 were pressed.

$\overline{\text{CH2}}$ RAM IC20

where switches of CE1 17 ~ 32 were pressed.

Conditions

Storage oscilloscope was used. This pulse occurs only when the internal voice switch is turned on. It can be observed with normal oscilloscope.



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
0.1mS TIME/DIV	
AC·DC	AC·DC

Probe 10 : 1

Where panel switch [1] was pressed.

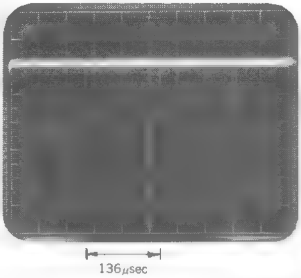
V-1 OUTPORT CHIP SELECT

Check Point

IC34 (8pin) $\overline{\text{LCDCS}}$

Conditions

Scanning situation where no panel switches were pressed.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
50μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

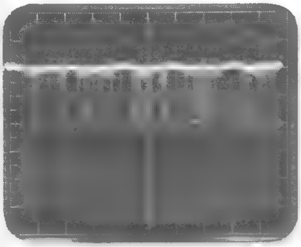
V-2 CHIP SELECT

Check Point

IC34 (8pin) $\overline{\text{LCDCS}}$

Conditions

Storage oscilloscope was used. A panel switch was pressed in the above picture (V–1 OUTPORT CHIP SELECT). The pulse can also be observed with normal oscilloscope.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
50μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

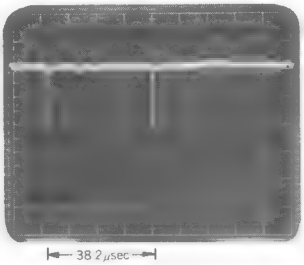
V-3 CHIP SELECT

Check Point

IC61 (3pin) $\overline{\text{ACEPT}}$

Conditions

This signal shows the acknowledge signal for the interrupt signal from the SUB. The pulse can also be observed with normal oscilloscope.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

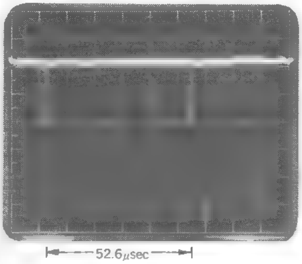
V-4 CHIP SELECT

Check Point

IC61 (3pin) $\overline{\text{ACEPT}}$

Conditions

This shows the same picture as the above one (V–3 CHIP SELECT) except that the DATA ENTRY control was operated and a number of keys were depressed at the same time.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10μS TIME DIV	
AC·DC	AC·DC

Probe 10 : 1

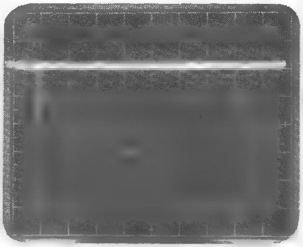
V-5 CHIP SELECT

Check Point

IC27 (6pin) $\overline{\text{DACCE}}$

Conditions

Storage oscilloscope was used because this signal is generated only on power on.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
50μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

V-6 CHIP SELECT

Check Point

CH1 IC26 (11pin)

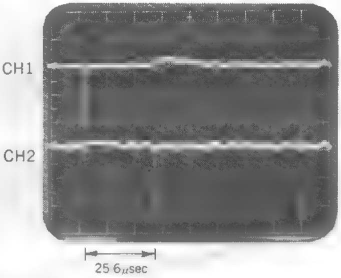
CH2 IC25 (11pin)

7-segment LED

data latching pulse

Conditions

Storage oscilloscope was used. This signal can be generated only when the 1 ~ 32 switches are turned on. The pulse can also be observed with a normal oscilloscope.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

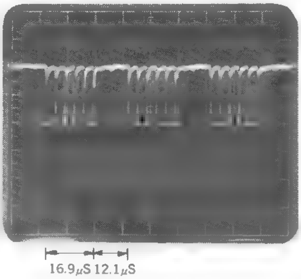
V-7 CHIP SELECT

Check Point

IC63 $\overline{\text{Y4}}$ (11pin) $\overline{\text{CRTE}}$

Conditions

Storage oscilloscope was used. This signal can be generated only when switches 1 ~ 32 are turned on. This pulse can also be observed with a normal oscilloscope.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

V-8 CHIP SELECT

Check Point

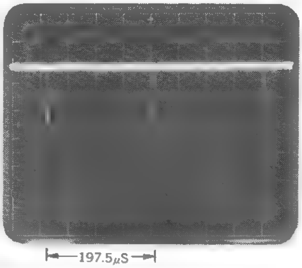
IC24 $\overline{\text{Y7}}$ (7pin)

Enable pulse for 7-segment LED

The picture on the right can be taken with this pulse trigger by E.

Conditions

Storage oscilloscope was used. This signal can be generated only when switches possibly enabling LED to change display were pressed. The pulse can also be observed with a normal oscillocope.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
50μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

VI-1 EG/OP

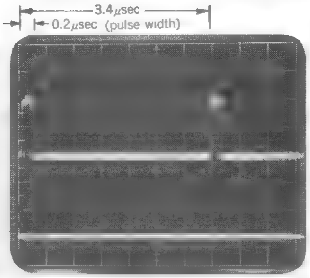
Check Point

EGS/OPS

KON data

Conditions

This situation occurs only when one voice is being pressed.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.5μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

VI-2 EG/OP

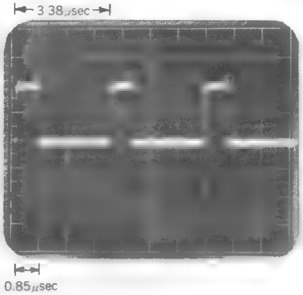
Check Point

EGS/OPS

KON data

Conditions

This situation shows that four notes starting from "Do" in sequence are being depressed.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
1μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

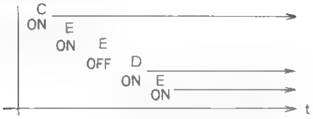
VI-3 EG/OP

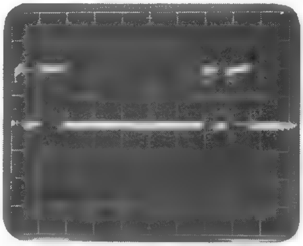
Check Point

EGS/OPS

KON data

Conditions





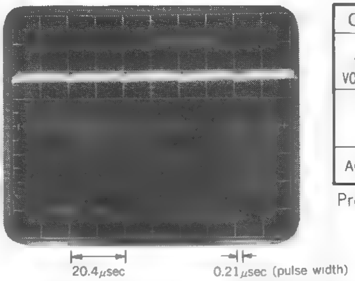
CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
0.5μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

VI-4 EG/OP

Check Point

$\overline{92Y96}$ IC36 (11pin)



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC·DC

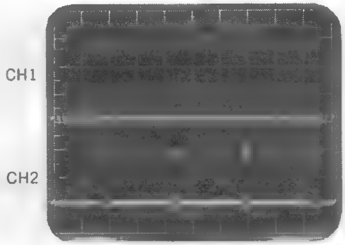
Probe 10 : 1

VI-5 EG/OP

Check Point

CH1 φEβ or Oβ

CH2 SH1 IC35 (11pin)



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
0.1μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

One period at every 12clocks

VI-6 EGS

Check Point

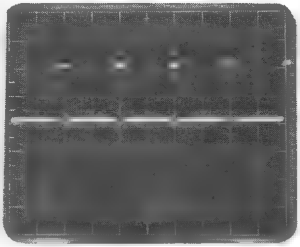
F1 (23pin)

Conditions

INTERNAL VOICE 1 = C3

Transpose MIDD C = C3

Turning on a key 16 times will result in the rectangular waveform as shown in the right picture. Depending on the musical interval, duty cycle varies. For each of F1 ~ F14, terminals are available, whose voltage level stay at +5V, 0V, or rectangular waveform.



CH.1	CH.2
0.2V VOLT DIV	VOLT DIV
10μS TIME DIV	
AC [✓ DC]	AC·DC

Probe 10 : 1

VI-7 EGS

Check Point

CH1 EC1 (41pin)

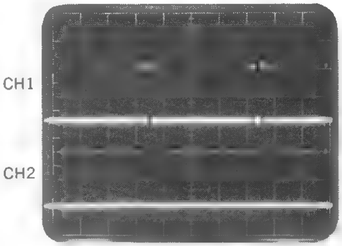
CH2 EC3 (43pin)

Conditions

INTERNAL VOICE 1

C3 key ON

Pattern example



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-8 EGS

Check Point

CH1 EC5 (45pin)

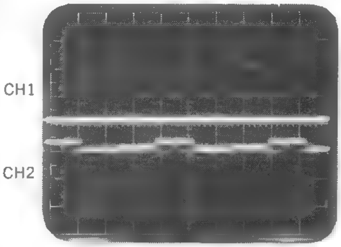
CH2 EC7 (47pin)

Conditions

INTERNAL VOICE 1

C3 key ON

Pattern example



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-9 EGS

Check Point

CH1 EC 9 (51pin)

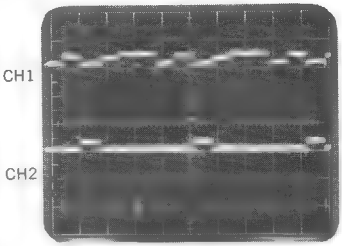
CH2 EC11 (53pin)

Conditions

INTERNAL VOICE 1

C3 key ON

Pattern example



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-10 EGS

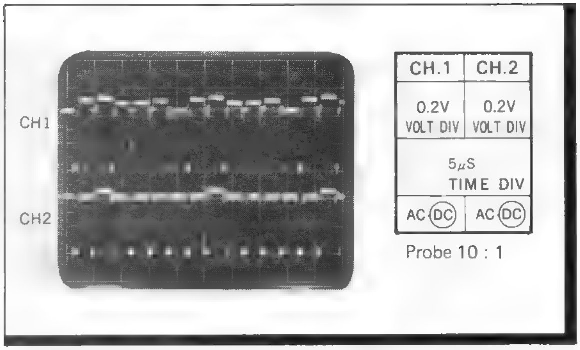
Check Point

CH1	EC9 (51pin)
CH2	EC11 (53pin)

Conditions

INTERNAL VOICE 1

Four keys C3, D3, E3, and F3 are depressed.



CH.1	CH.2
0.2V VOLT DIV	0.2V VOLT DIV
5μS TIME DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-11 OPS

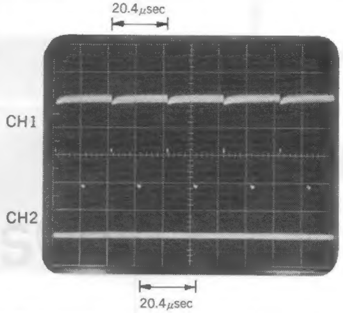
Check Point

CH1 $\overline{92Y96}$ IC36 (11pin)

CH2 SF0 IC36 (39pin)

Conditions

This signal is generated when one voice key is turned on (INTERNAL VOICE 1 or 17 C3 key ON).
This allows compressed digital data to be expanded again. Hence, with attenuating sound, this signal appears at the leading edge and then disappears. With modulated sound, this signal can be observed to move sideways within a range of 1 ms on the oscilloscope in synchronization with the modulation period.



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
10μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-12 OPS

Check Point

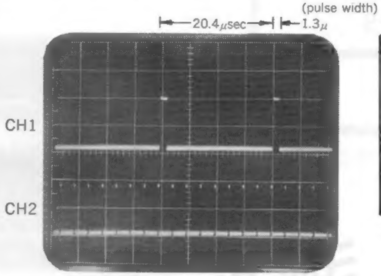
CH1 SF0 IC36 39pin

CH2 SH2 IC36 10pin

Conditions

Only one voice key is on.

This signal can be observed using a continuous sound.



Output between SH1 and SH2

CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
5μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-13 OPS

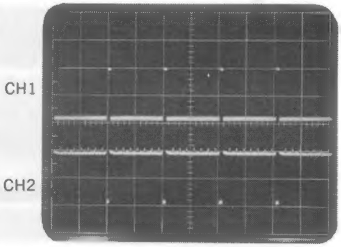
Check Point

CH1 SF0 1, 2 IC36 (39, 40, 41 pin)

CH2 SF2 IC36 (42pin)

Conditions

INTERNAL VOICE 17 C3 key ON
The signal can be observed within a range of 10 μS ⟶ 0.5 mS. When the modulation is on, this signal is seen to move sideways on the oscilloscope.



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
10μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-14 OPS

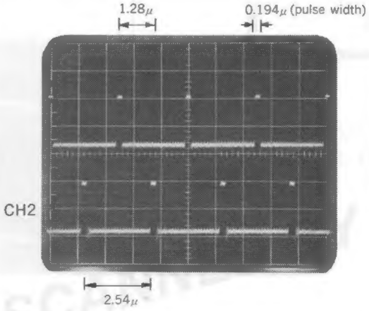
Check Point

CH1 SH1 IC36 (9pin)

CH2 SH2 IC36 (19pin)

Conditions

This signal can be constantly observed whether the key is on or off.



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
1μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

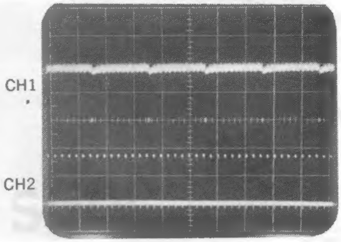
Probe 10 : 1

VI-15 OPS

Check Point

CH1 $\overline{92Y96}$

CH2 SH1 (9pin)



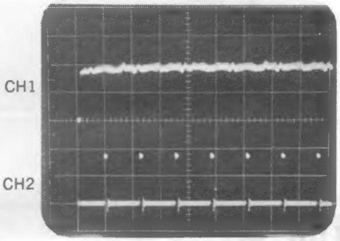
CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
10μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-16 OPS

Check Point

This shows an expanded one of the above picture (VI–15 OPS).



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
2μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

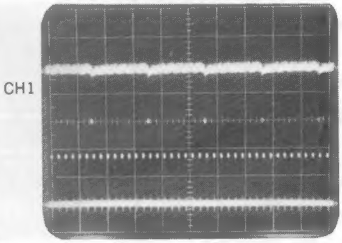
Probe 10 : 1

VI-17 OPS

Check Point

CH1 $\overline{92Y96}$

CH2 SH2 (10pin)



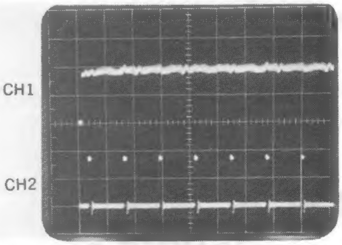
CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
10μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VI-18 OPS

Check Point

The right shows the expanded one of the above picture (VI–17 OPS).



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
2μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

Probe 10 : 1

VII-1 D/A

Check Point

CH1 D/A (1, 3, 5, 7, 9, 11pin)

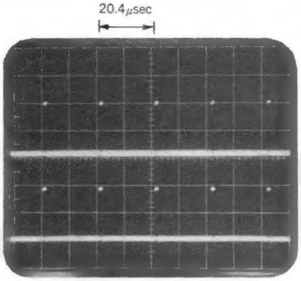
CH2 D/A (2, 4, 6, 8, 10, 12pin)

Conditions

INTERNAL VOICE 1

C3 key ON

The sampling period can be observed. Actual change of bit pattern can be seen within a range of 5 ms on the oscilloscope.



CH.1	CH.2
0.2V VOLT/DIV	0.2V VOLT/DIV
10μS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

VII-2 D/A

Check Point

CH1 D/A (9pin)

CH2 D/A (10pin)

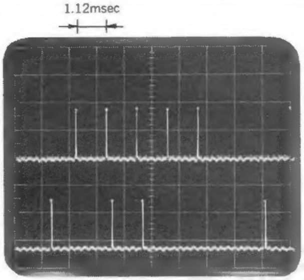
Conditions

Storage oscilloscope was used.

INTERNAL VOICE 3

C3 key ON

The sweeping time was changed from 10 μs to 1mS (TIME/DIV) in the above picture(**VII**–1 D/A).The change can also be observed using an storage oscilloscope.



About 2.2msec. (pulse width)

CH.1	CH.2
0.2V VOLT/DIV	VOLT/DIV
1mS TIME/DIV	
AC [✓ DC]	AC [✓ DC]

VII-3 D/A

Check Point

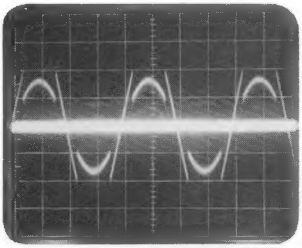
D/A OUT IC40 (6pin)

Conditions

INTERNAL VOICE 3

C3 key ON

Only OP1 is on (Other OPs are disabled).



CH.1	CH.2
20mV VOLT/DIV	/DIV
2mS TIME/DIV	
[✓ AC]·DC	AC·DC

The center is DC+7.5V.

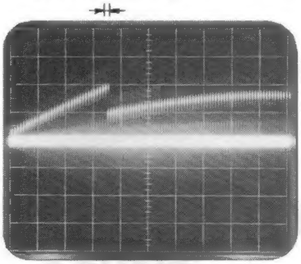
VII-4 D/A

Check Point

D/A OUT IC40 6pin

This shows the expanded one of the above picture (**VII**–3 D/A).

The gap between each word dat a is 20.4μsec.



CH.1	CH.2
20mV VOLT/DIV	/DIV
0.2mS TIME/DIV	
[✓ AC] DC	AC·DC

YAMAHA